CSCE A405/A605 (Adv) Artificial Intelligence

Informed Search

Ref: Artificial Intelligence: A Modern Approach, 4th ed by Stuart Russell and Peter Norvig, chapter 3

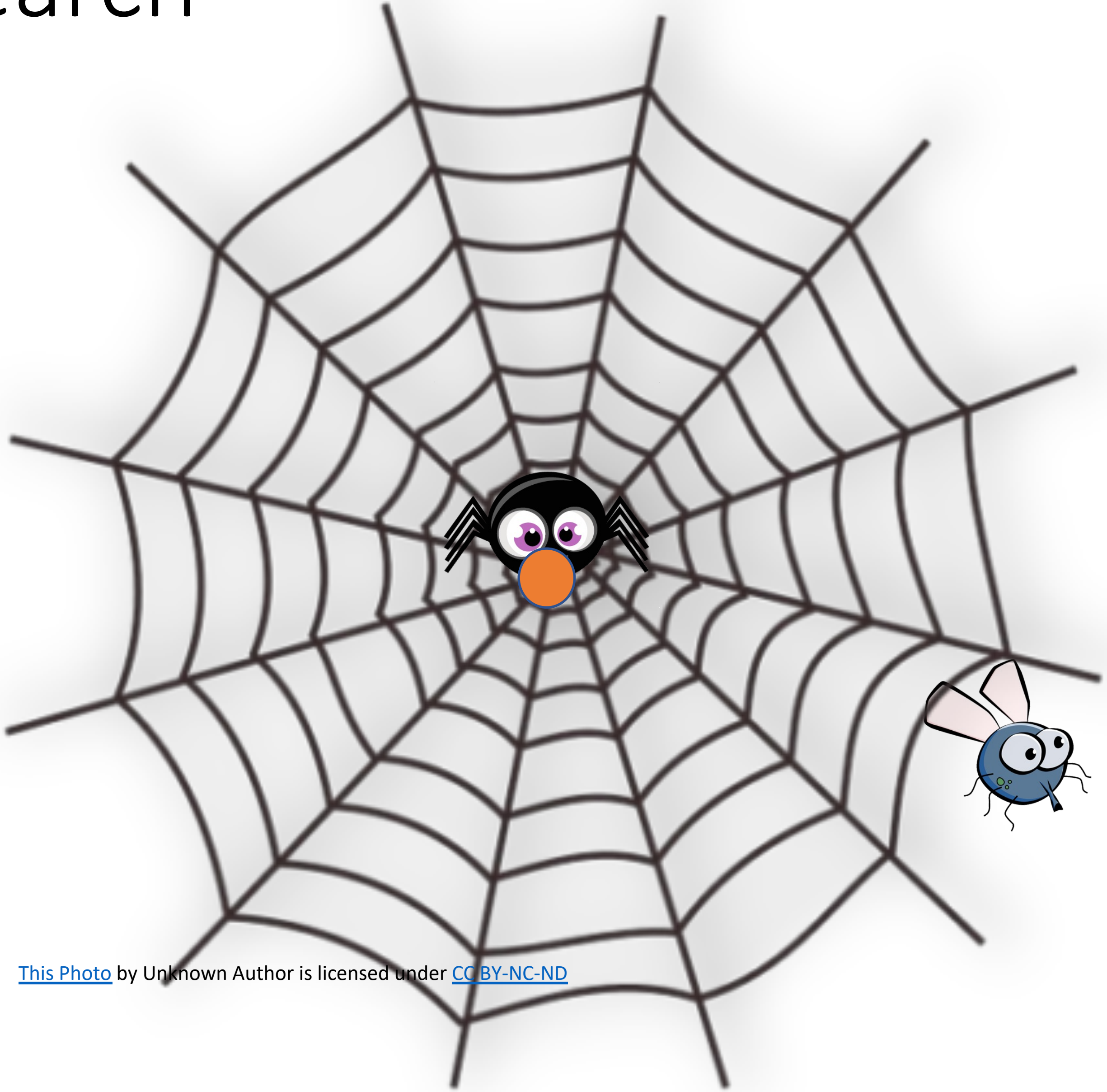
Instructor: Masoumeh Heidari (mheidari2@Alaska.edu)

UAA College of Engineering
UNIVERSITY of ALASKA ANCHORAGE

Learning objective

- Greedy best-first search
- A* search

Informed search

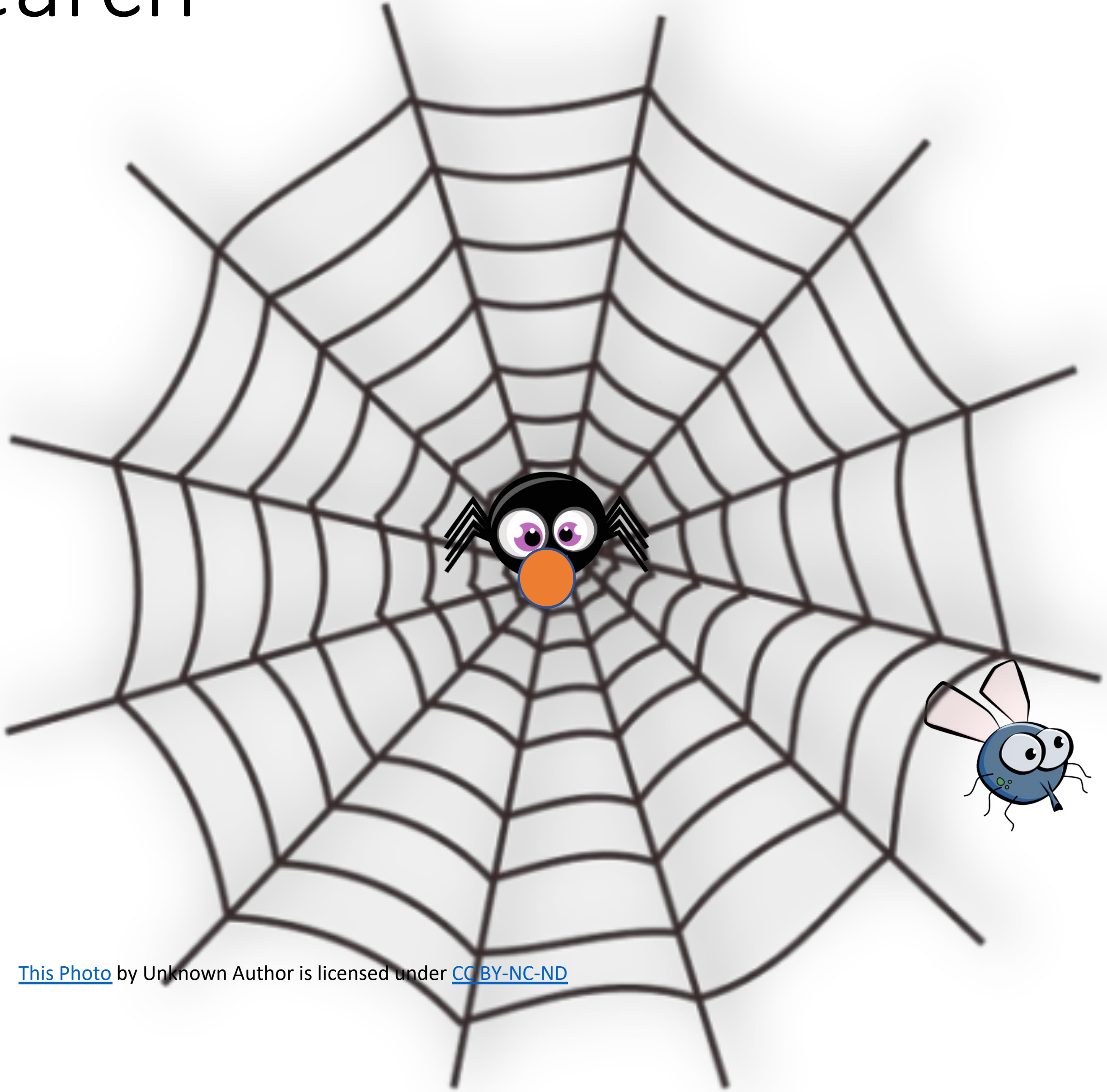


This Photo by Unknown Author is licensed under CC BY-NC

This Photo by Unknown Author is licensed under CC BY-NC-ND

UAA College of Engineering
University of Alaska Anchorage

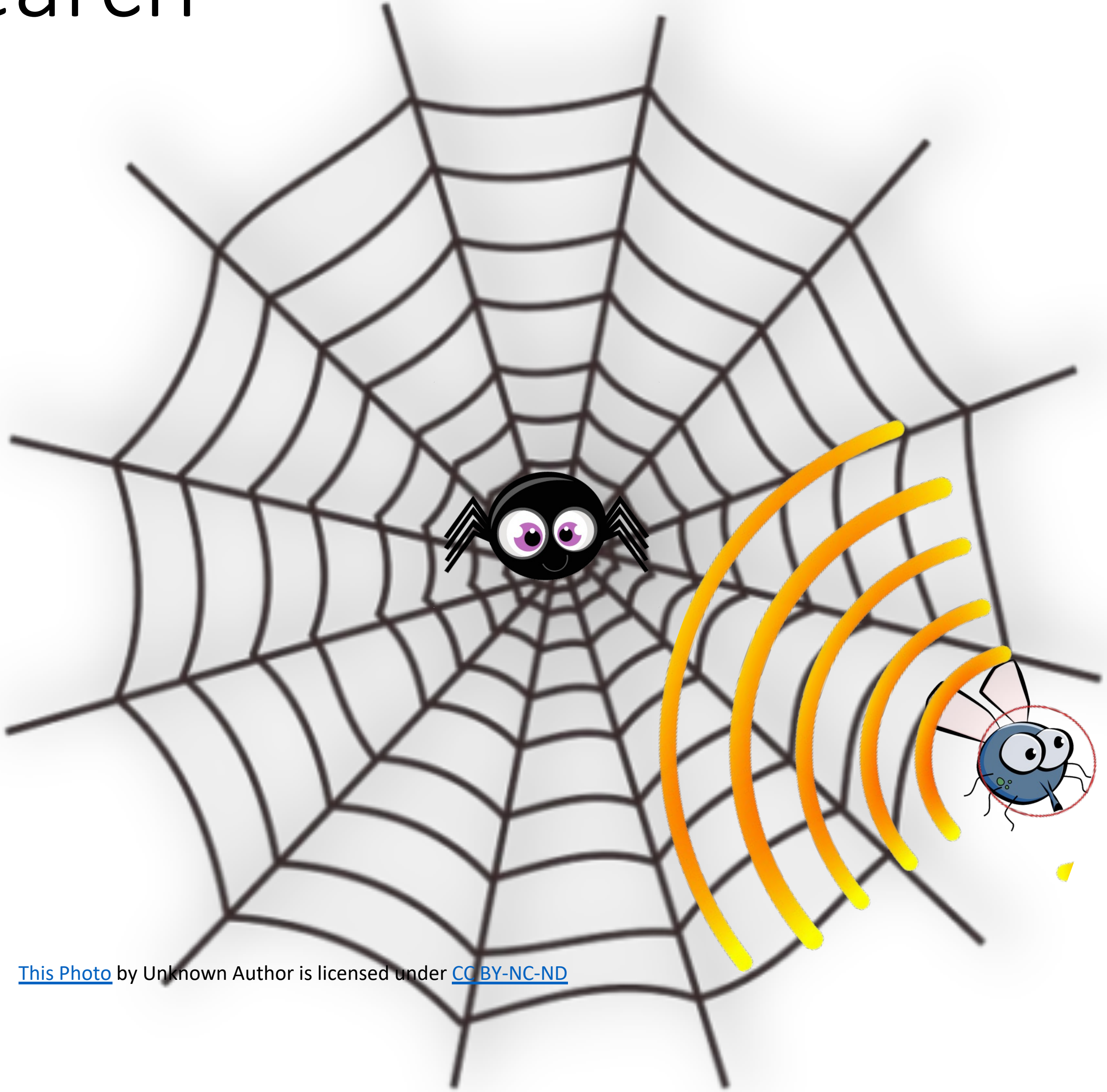
Informed search



This Photo by Unknown Author is licensed under CC BY-NC

This Photo by Unknown Author is licensed under CC BY-NC-ND

Informed search



[This Photo](#) by Unknown Author is licensed under [CC BY-NC](#)

[This Photo](#) by Unknown Author is licensed under [CC BY-NC-ND](#)

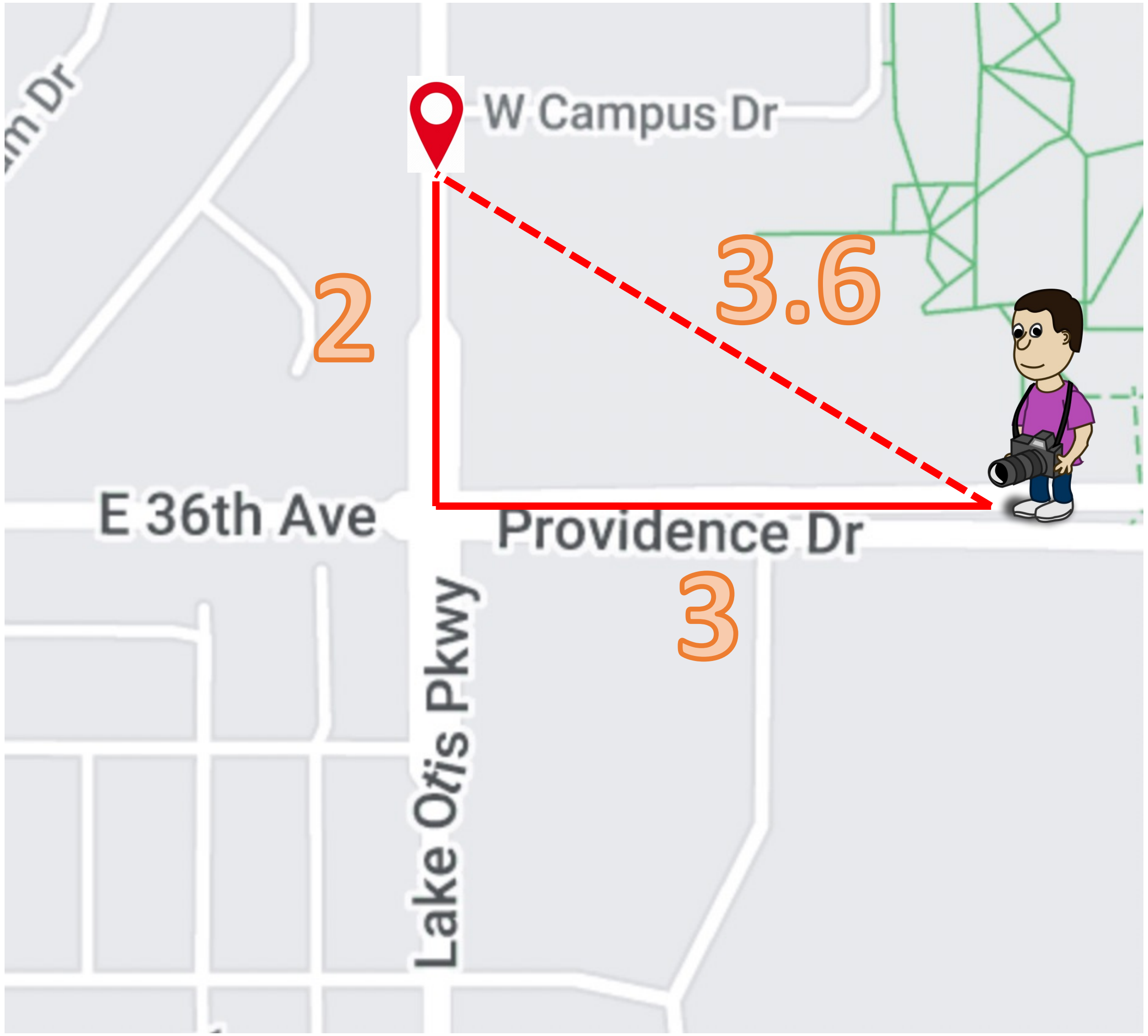
Informed search

- Uses domain-specific hints about the location of goals—can find solutions more efficiently than an uninformed strategy. The hints come in the form of a heuristic function, denoted h(n).

h (n) = ***Estimated*** cost of the cheapest path from the state at node n to a goal state.

Informed search

For example, in route-finding problems, we can estimate the distance from the current state to a goal by computing the straight-line distance on the map between the two points.



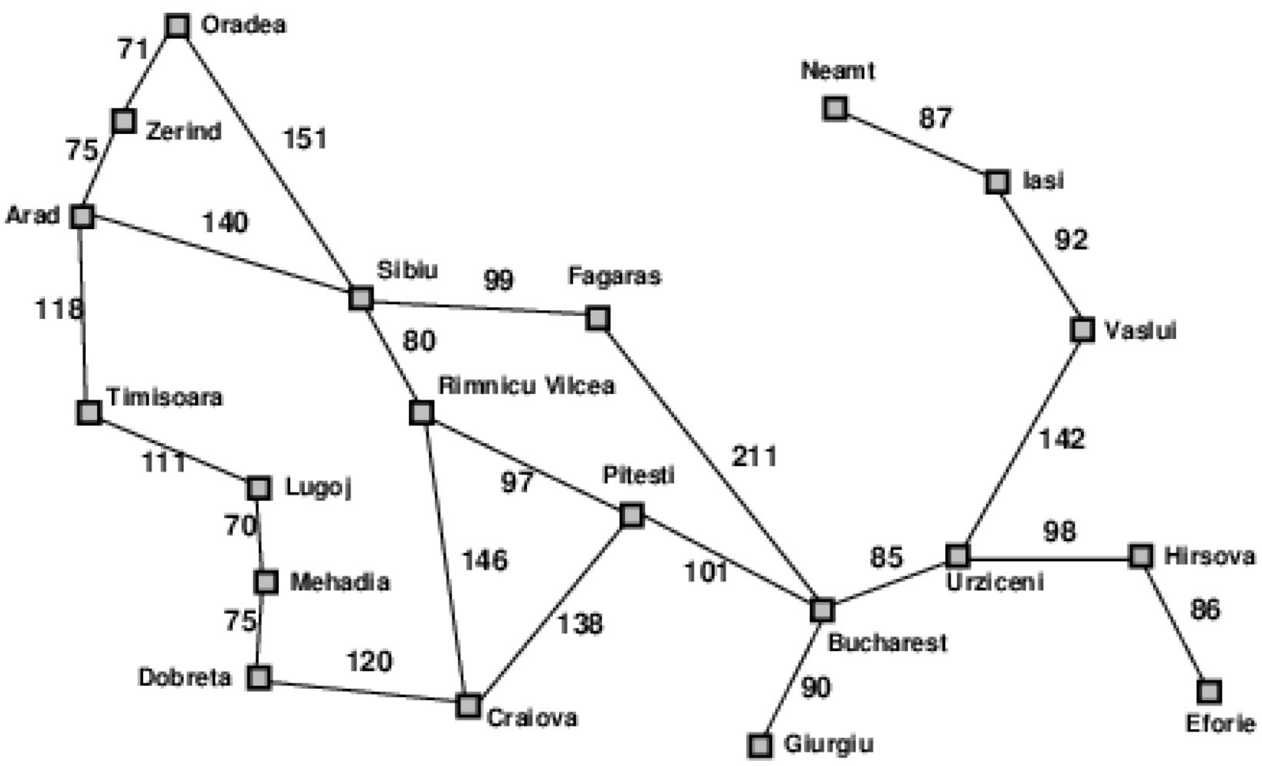
Greedy best-first search

- A form of best-first search that expands first the node with the lowest h(n) value the node that appears to be closest to the goal—on the grounds that this is likely to lead to a solution quickly.

- The evaluation function is f(n) = h(n).

Greedy best-first search

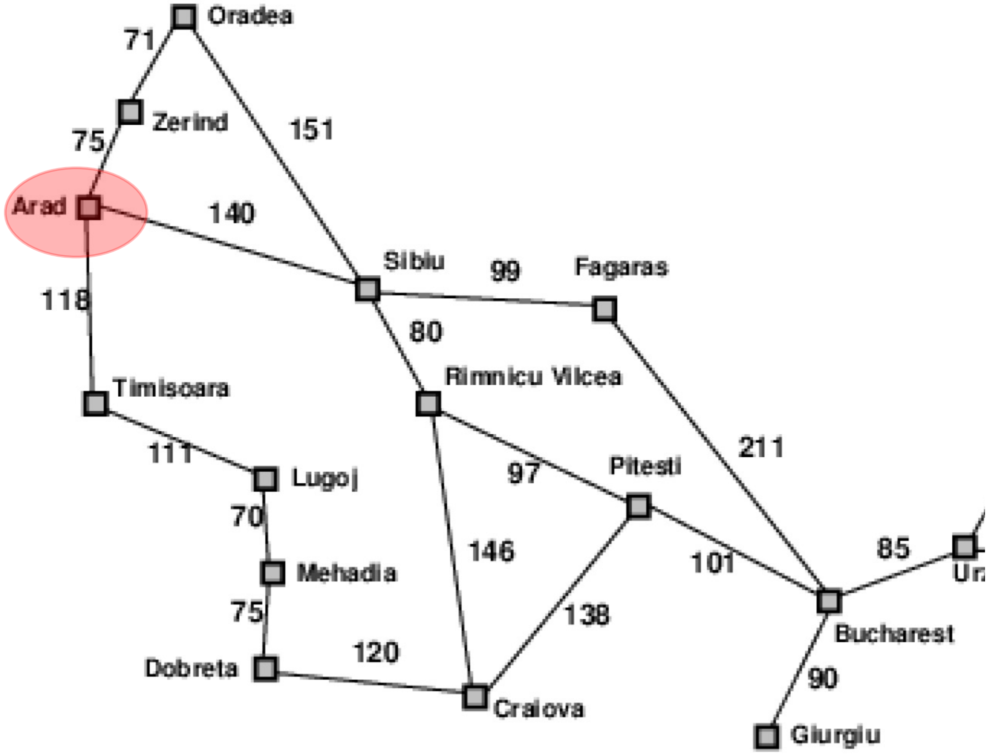
h(n)



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

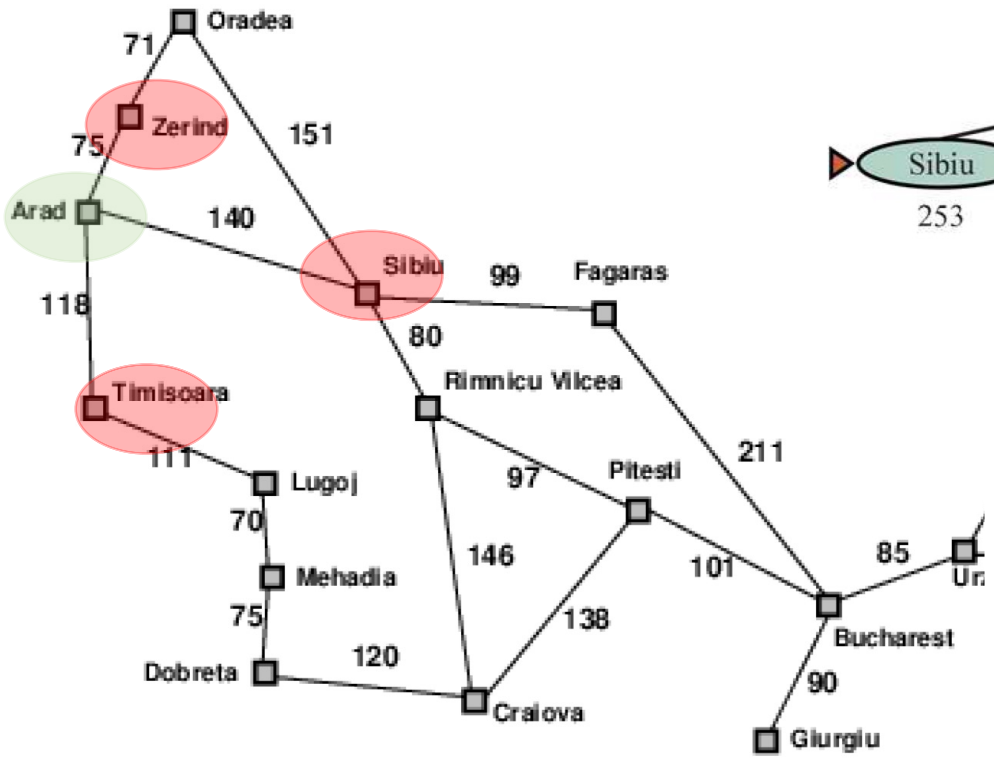
Greedy best-first search

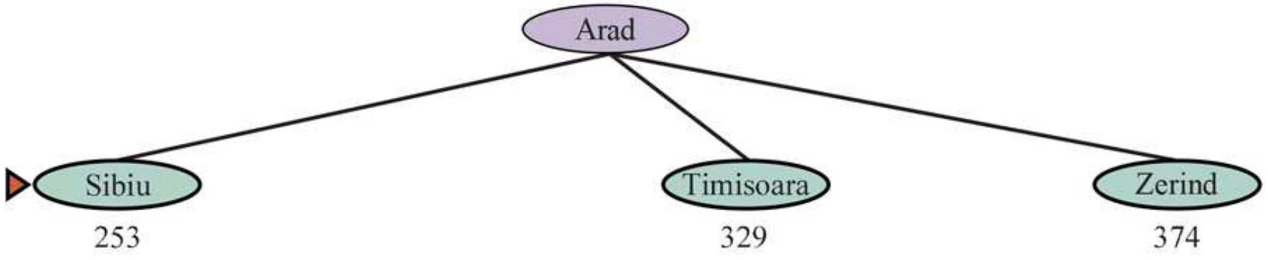


Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

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

Greedy best-first search

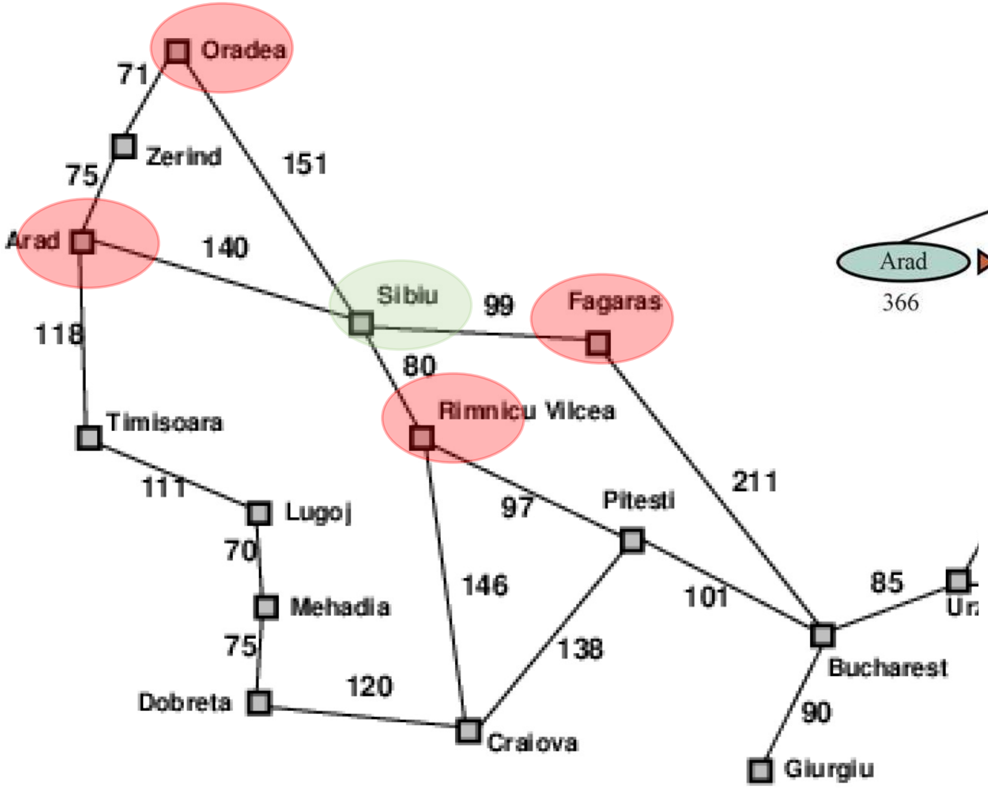


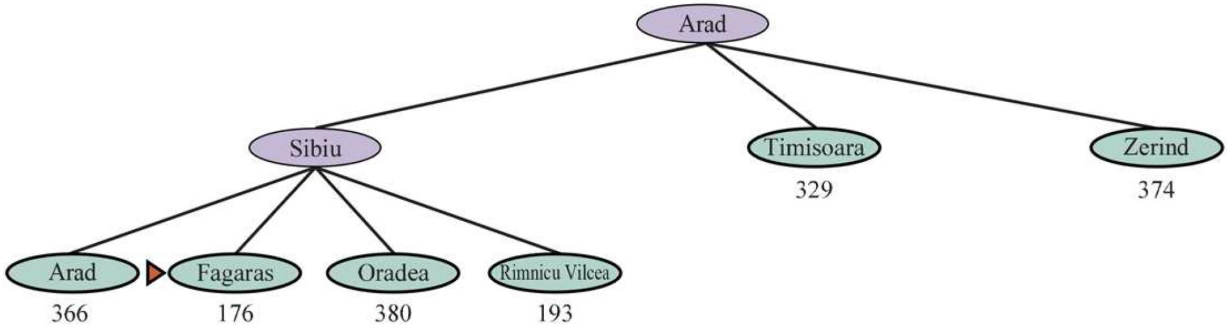


Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

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

Greedy best-first search

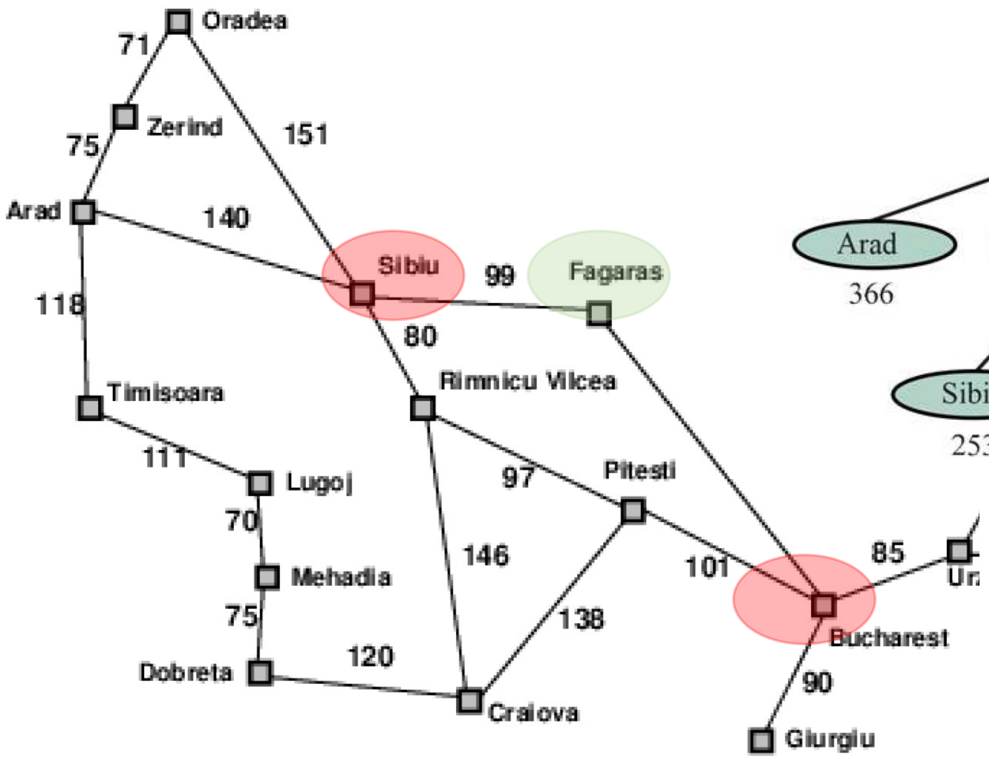


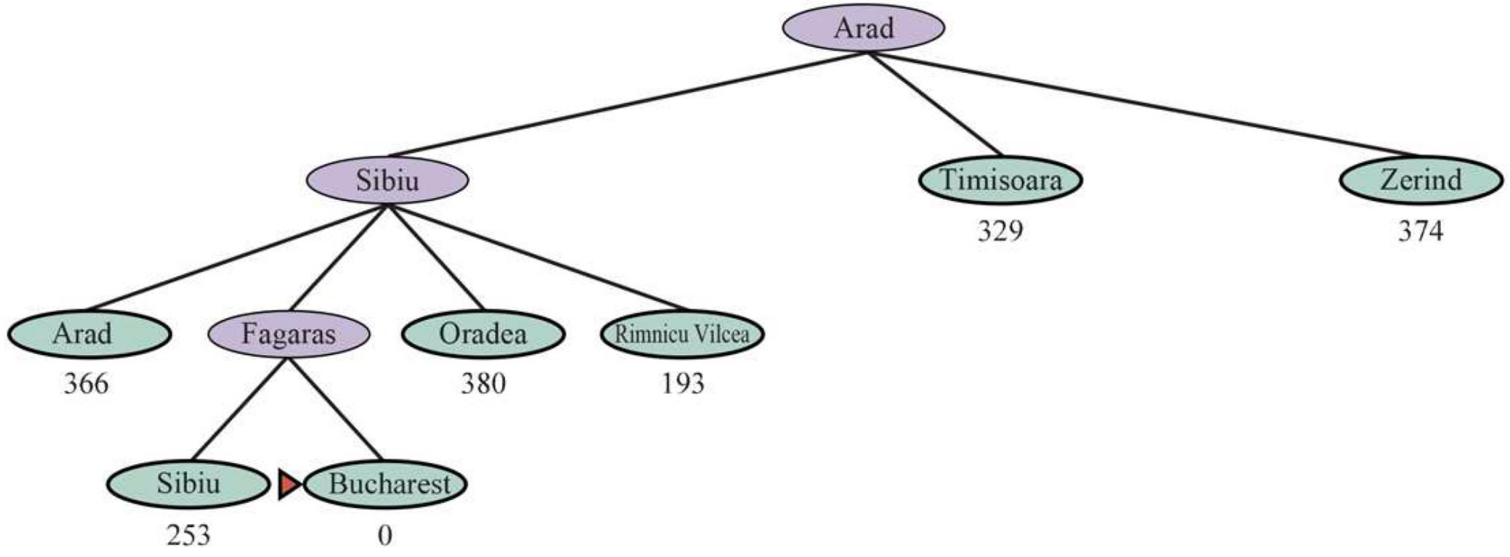


Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

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

Greedy best-first search

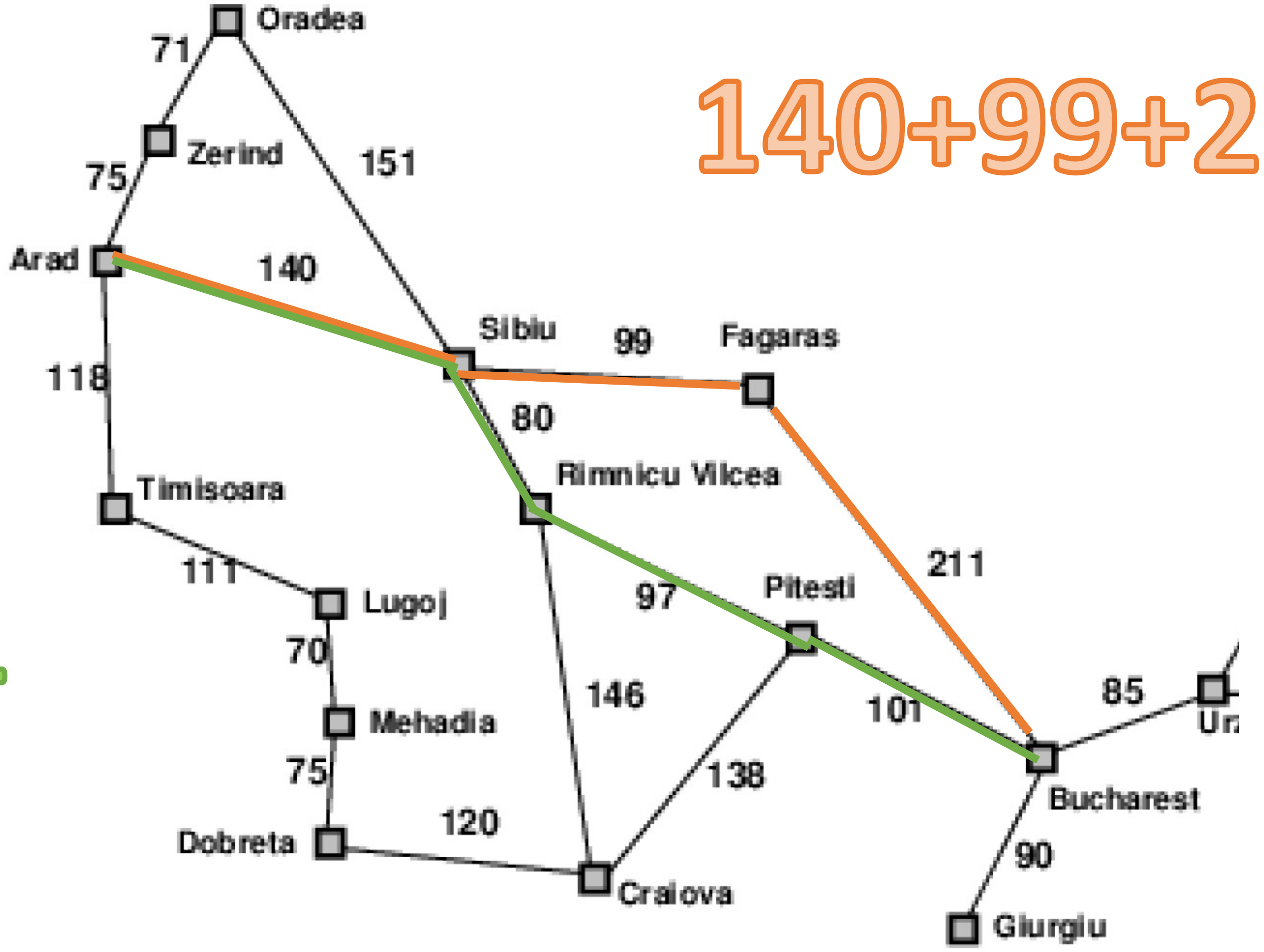




Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

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

Greedy best-first search



140+99+211=450

140+80+97+
101 = 418

A* search

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

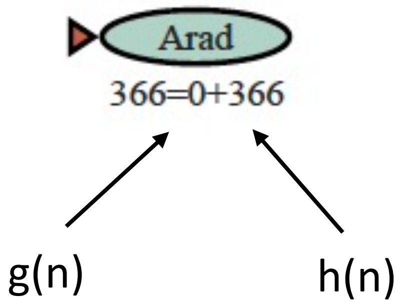
Best-first search

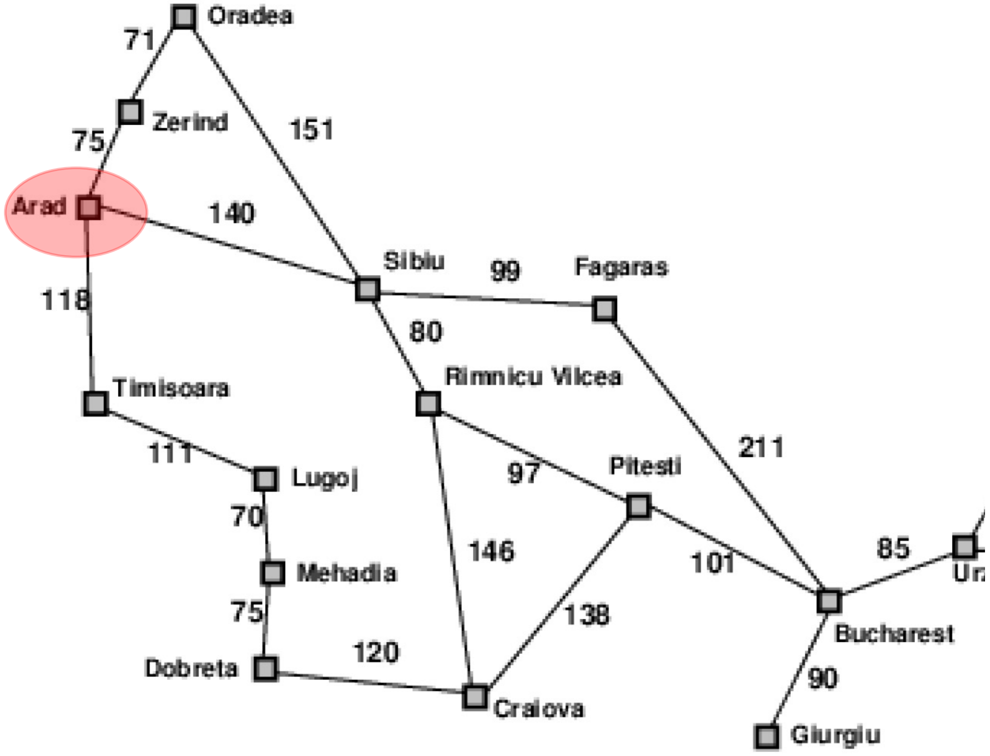
Greedy best-first search

A* search

- g(*n*) is the path cost from the initial state to node *n*.
- h(*n*) is the estimated cost of the shortest path from n to a goal state.
- f(*n*) is the estimated cost of the best path that continues from *n* to a goal.

A* search

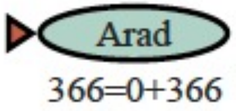


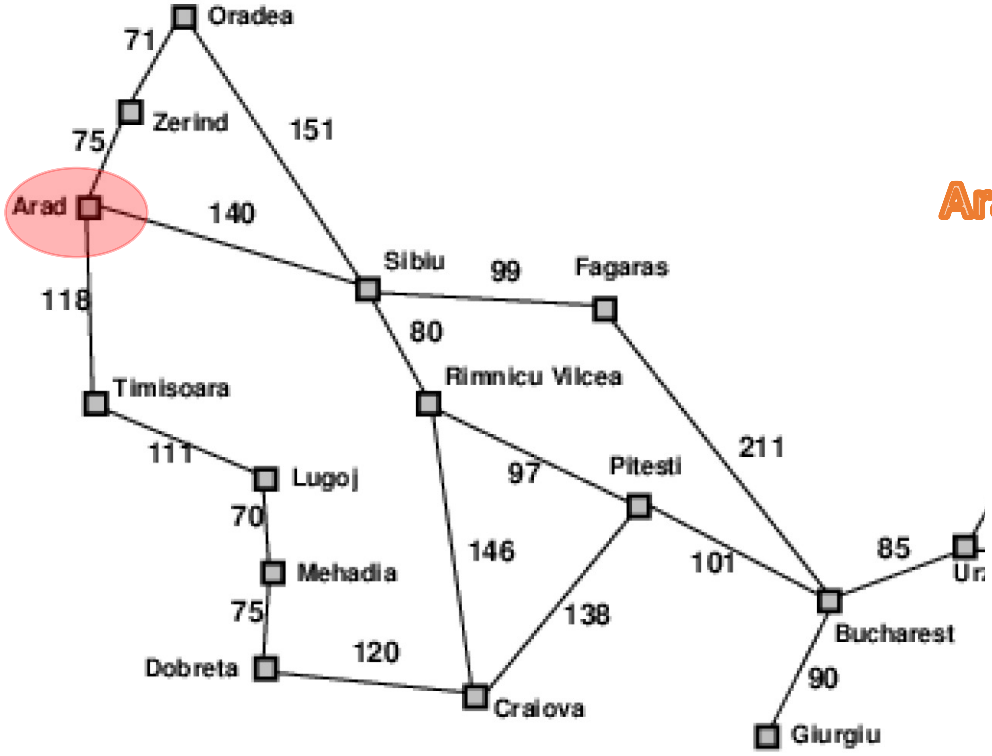


Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

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

A* search



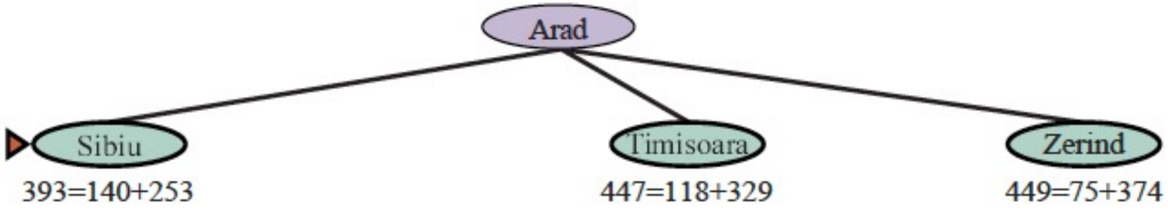


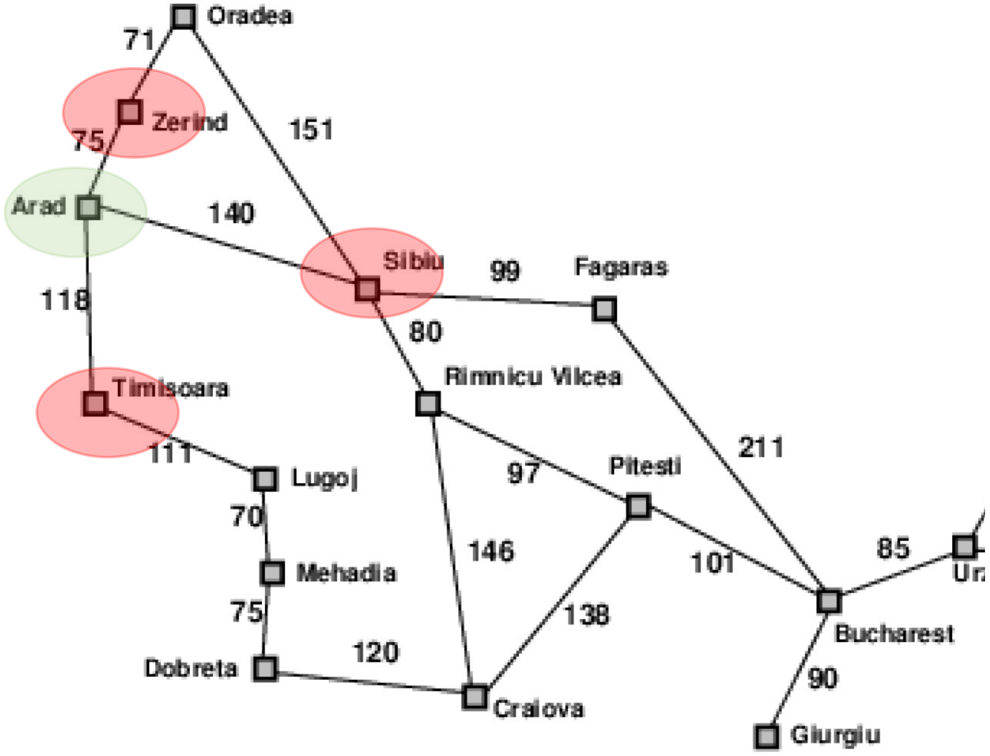
Arad-> Sibiu -> Rimnicu Vilcea -> Pitesti -> Bucharest

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

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

A* search

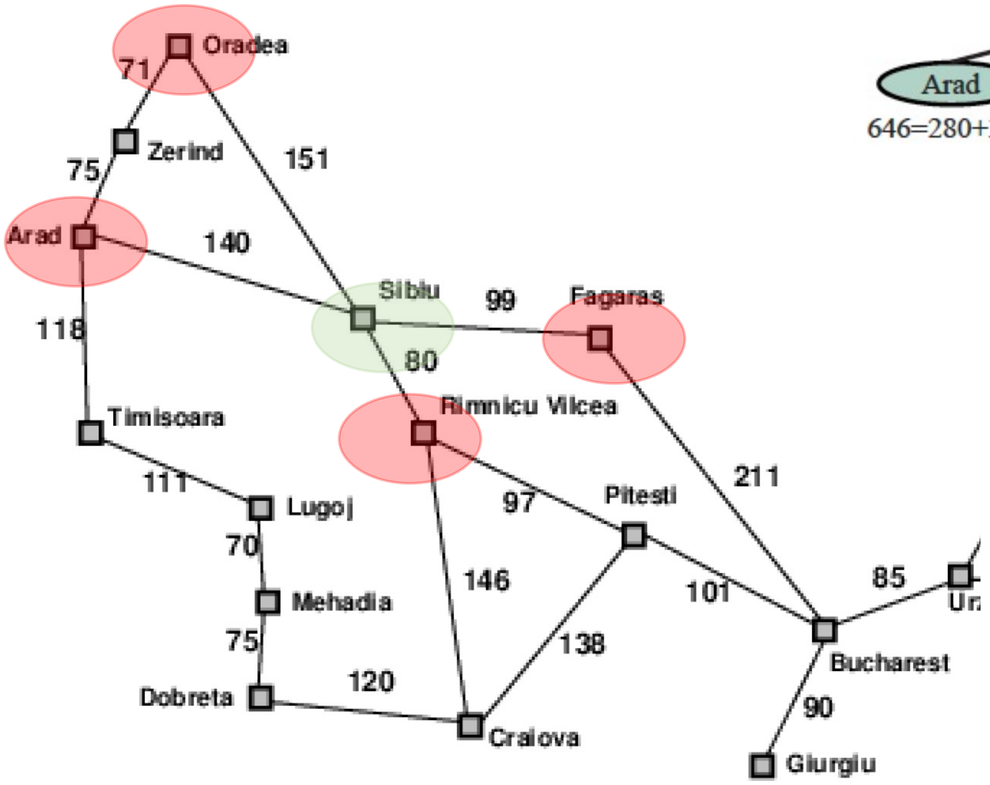


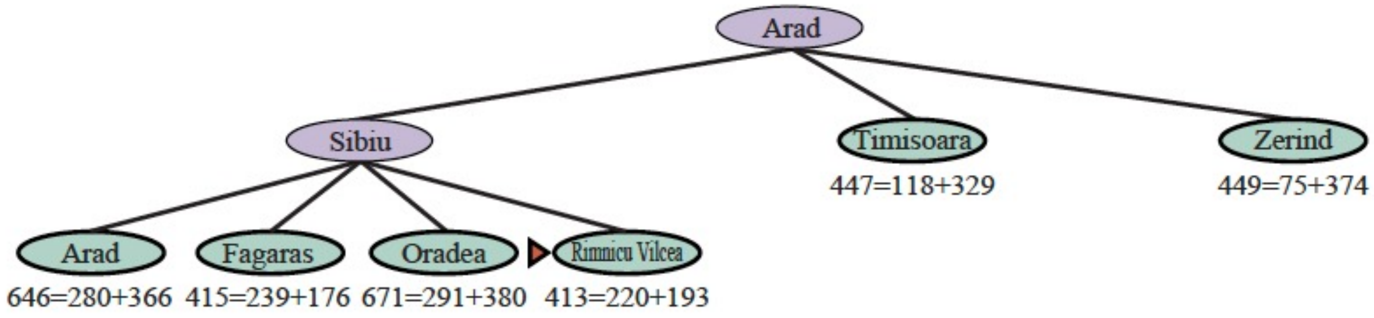


Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

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

A* search

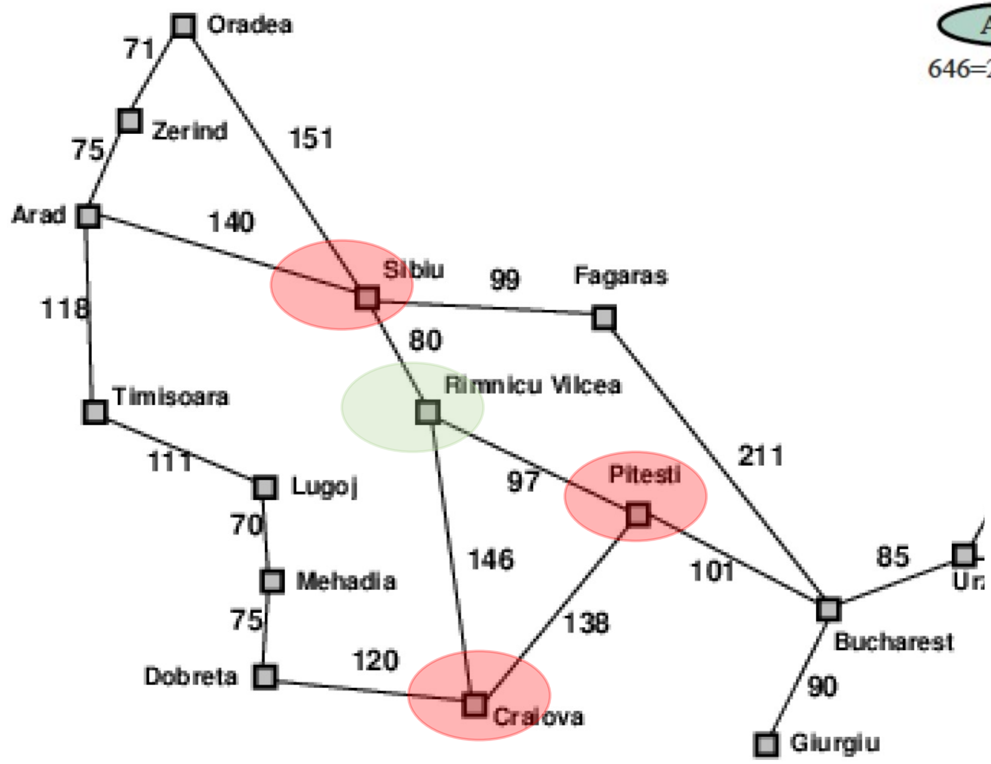


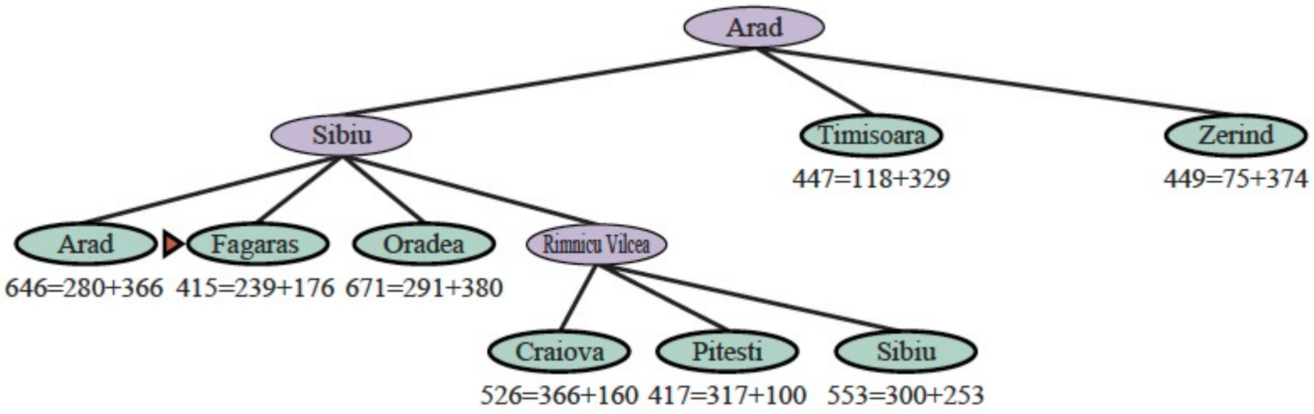


Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

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

A* search

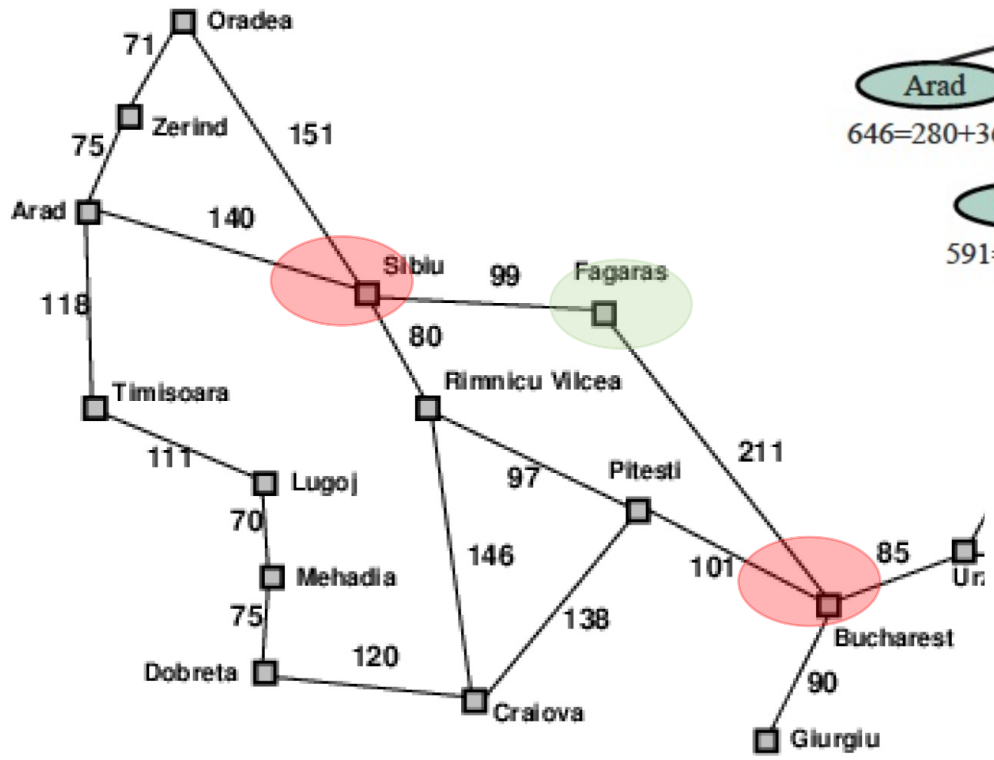


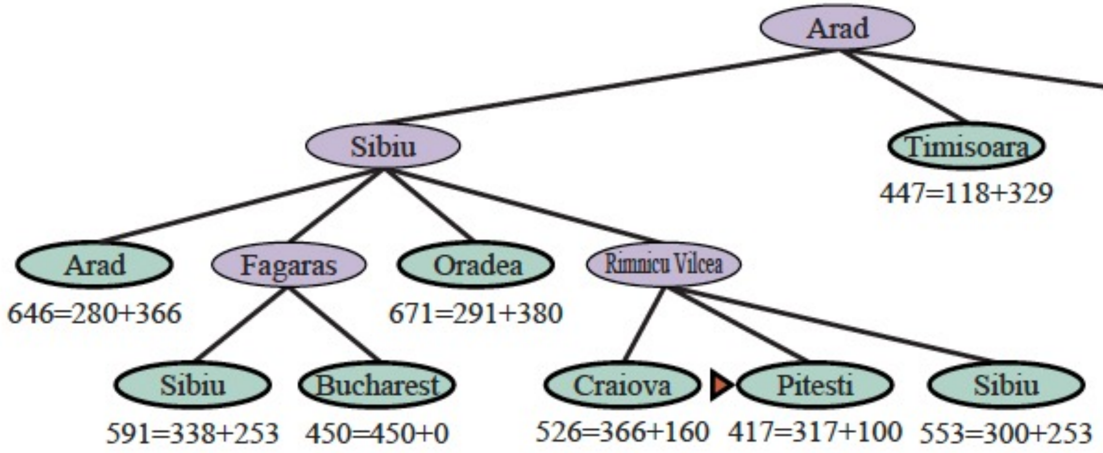


Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

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

A* search

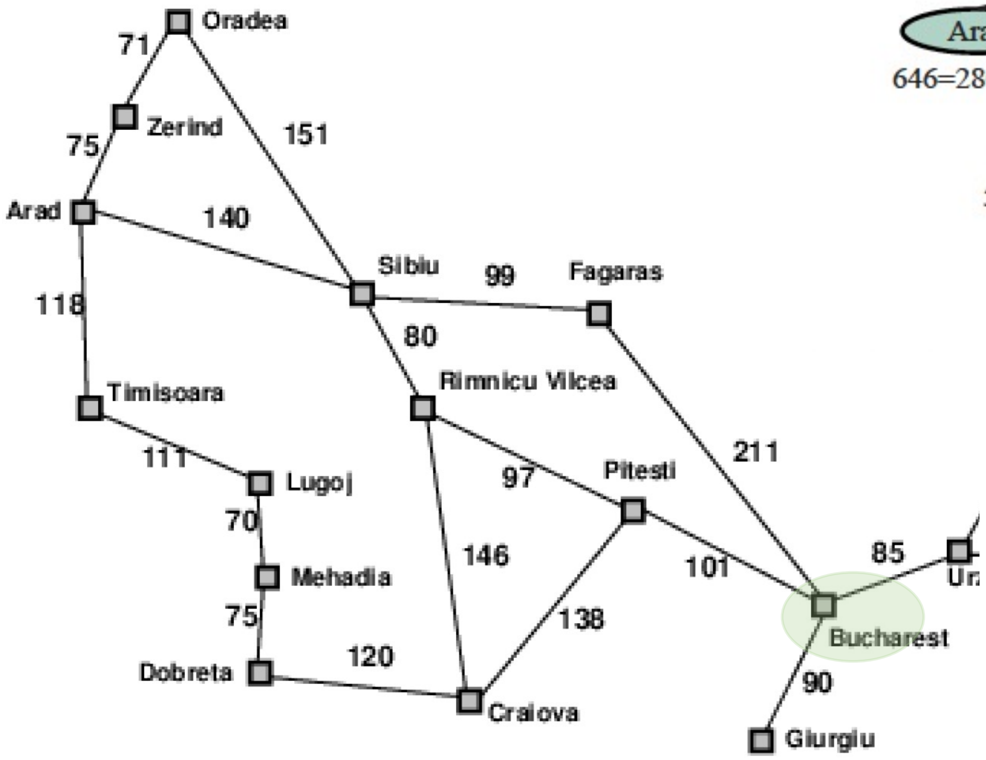


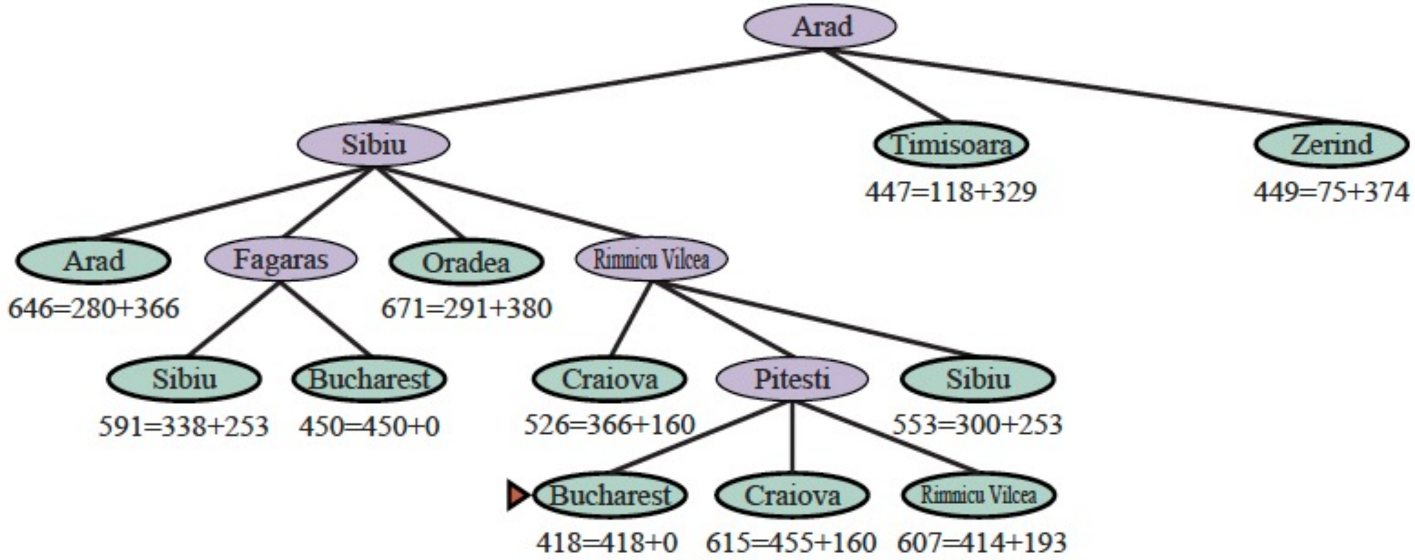


Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

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

A* 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

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

A* applications

- AlShawi, Imad S., et al. "Lifetime enhancement in wireless sensor networks using fuzzy approach and A-star algorithm." *IEEE Sensors journal* 12.10 (2012): 3010-3018.
- Tseng, Fan Hsun, et al. "A star search algorithm for civil UAV path planning with 3G communication." *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE, 2014.
- Zheng, Tao, Yanqiang Xu, and Da Zheng. "AGV path planning based on improved A-star algorithm." 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). IEEE, 2019.
- Wang, Chunbao, et al. "Path planning of automated guided vehicles based on improved A-Star algorithm." 2015 IEEE International Conference on Information and Automation. IEEE, 2015.
- Hu, Daqi, et al. "An Improved A-Star Algorithm for Path Planning of Outdoor Distribution Robots." Proceedings of the Asia Conference on Electrical, Power and Computer Engineering. 2022.
- Mathew, Geethu Elizebeth. "Direction based heuristic for path finding in video games." *Procedia Computer Science* 47 (2015): 262-271.

Properties of A*

- Assuming all action costs are positive, and the state space either has a solution or is finite, A* is complete.

- Cost-optimal?

- **Admissibility:** an admissible heuristic is one that never overestimate the cost to reach a goal.

- With an admissible heuristic, A* is cost-optimal.

Properties of A*

- Proof by contradiction:
  - Suppose the optimal path has cost C*, but the algorithm returns a path with cost C > C*.

Not cost-optimal

  - Then there must be some node *n* which is on the optimal path and is unexpanded.

Otherwise, we would have returned the optimal solution

  - Given $g^*(n)$ as the cost of the optimal path from the start to *n* and $h^*(n)$ as the cost of optimal path from *n* to the nearest goal:

Properties of A*

- $f(n) > C^*$

Otherwise, *n* would have been expanded.

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

- $f(n) = g^*(n) + h(n)$

Because *n* is on an optimal path.

- $f(n) \leq g^*(n) + h^*(n)$

Because of admissibility, $h(n) \leq h^*(n)$

- $f(n) \leq C^*$

The first and last lines form a contradiction -> A* returns only cost-optimal path.

Recap

- Greedy best-first search
  - Expands nodes with minimal *h(n).* It is not optimal but is often efficient.

- A* search
  - Expand nodes with minimal $f(n) = g(n) + h(n)$.
  - A* is complete and optimal, provided that *h(n)* is admissible.