Lecture 7
Informed Search

CS 180 – Intelligent Systems

Dr. Victor Chen

Spring 2021

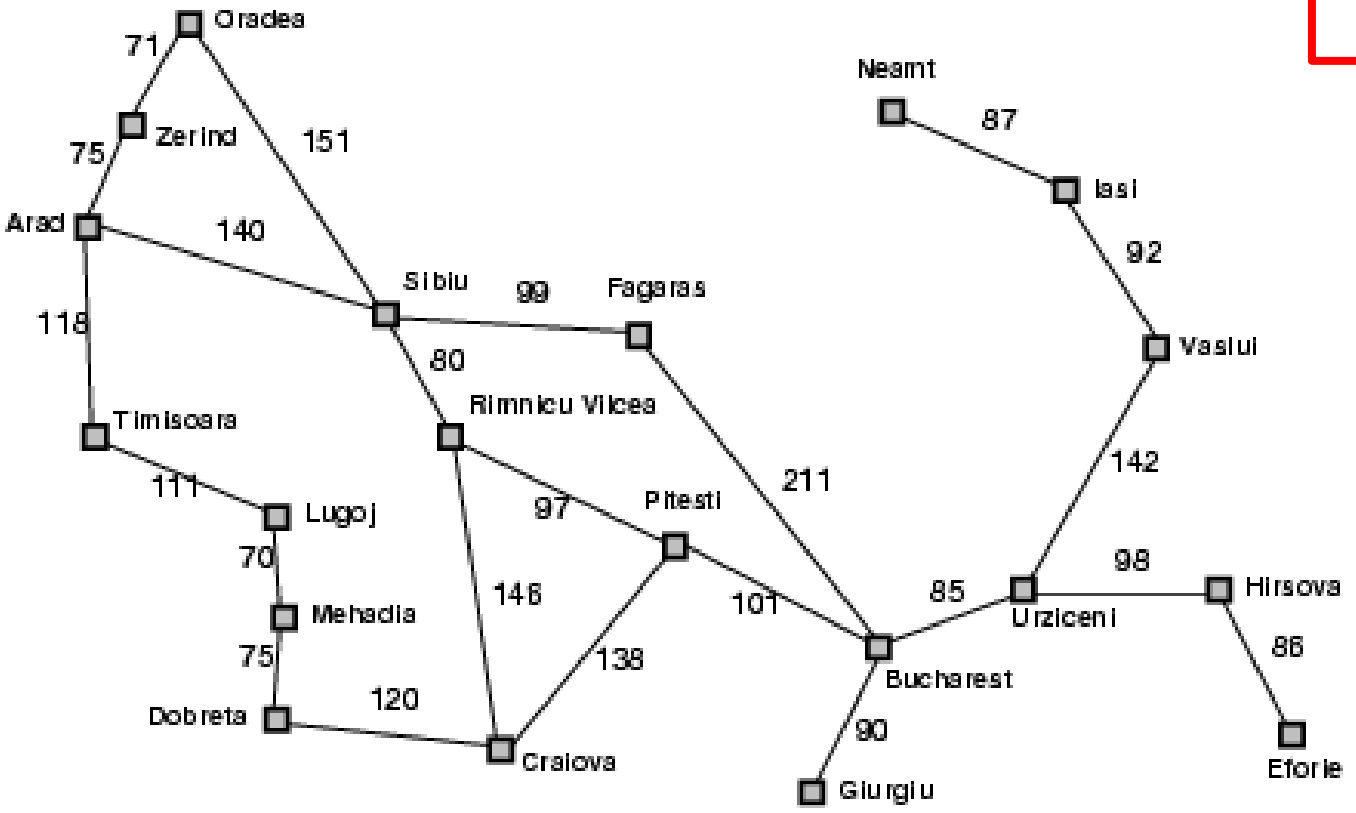
Informed Search

Informed (heuristic) Search

Idea: Use a ***heuristic function*** $h(n)$ to rank nodes in the *frontier* and expand the node with the lowest $h(n)$ value

- Greedy search (Best-first search)
- A* search

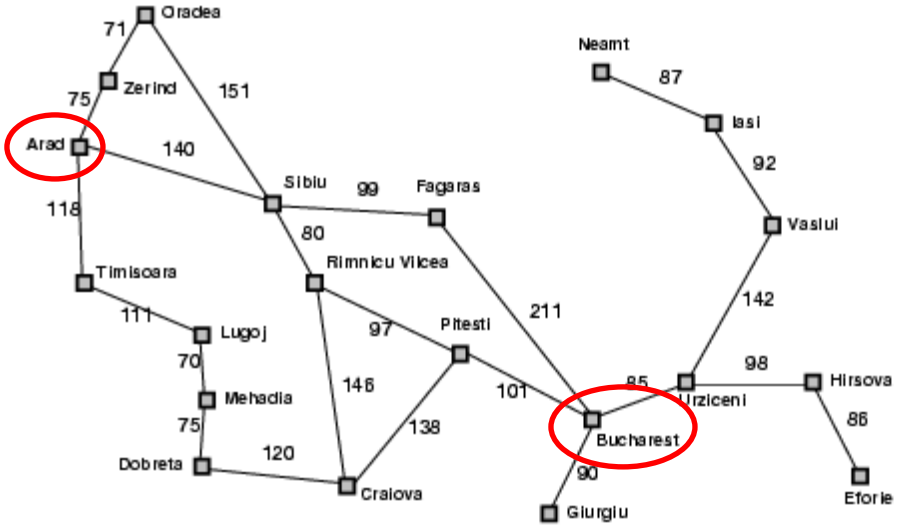
Heuristic for the Romania problem



heuristic function h(n)

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	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

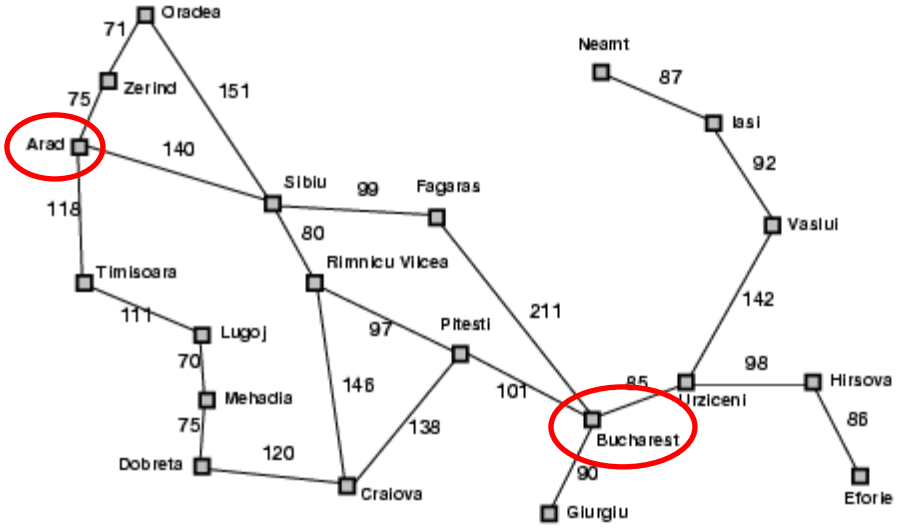
Greedy best-first search example



Straight-line distance to Bucharest

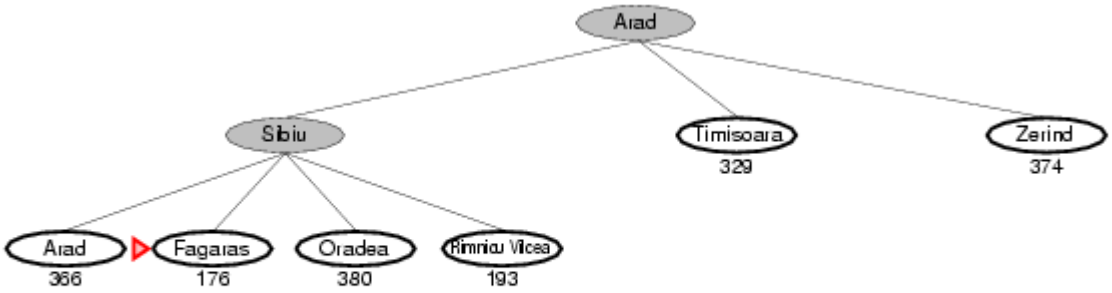
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

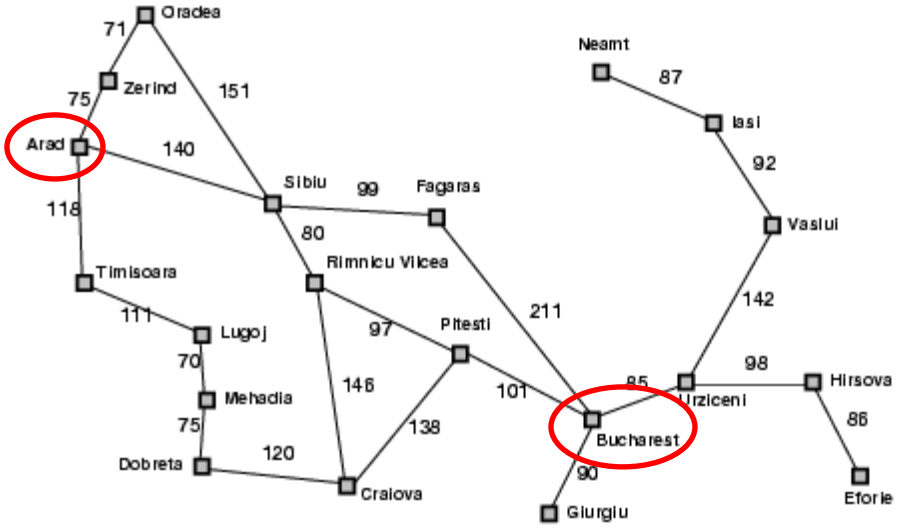
Greedy best-first search example



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	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

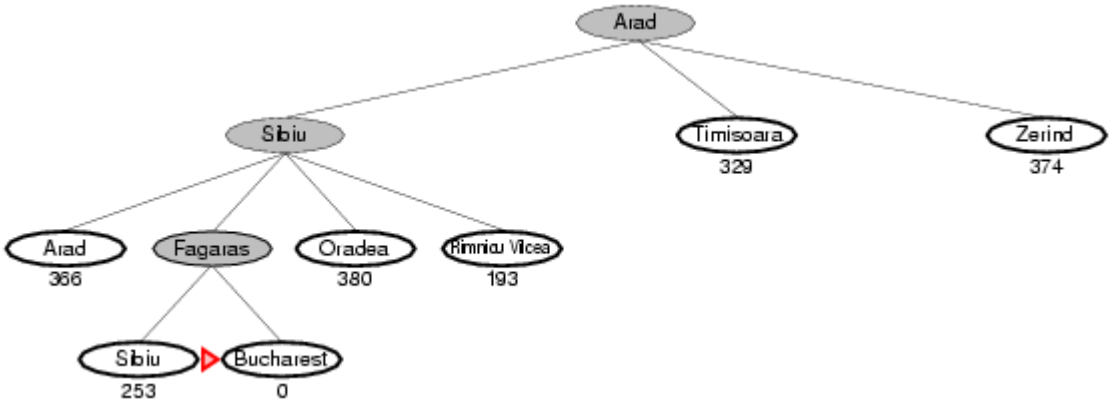
Greedy best-first search example

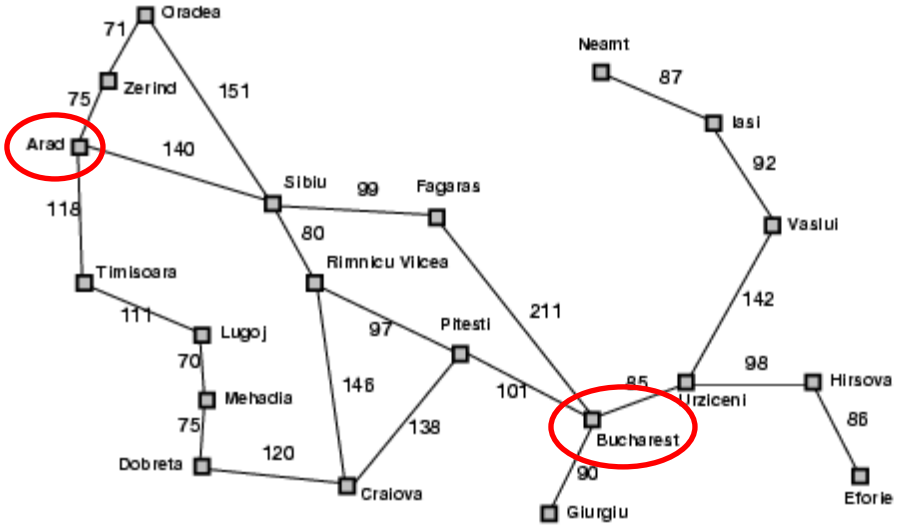




Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Greedy best-first search example



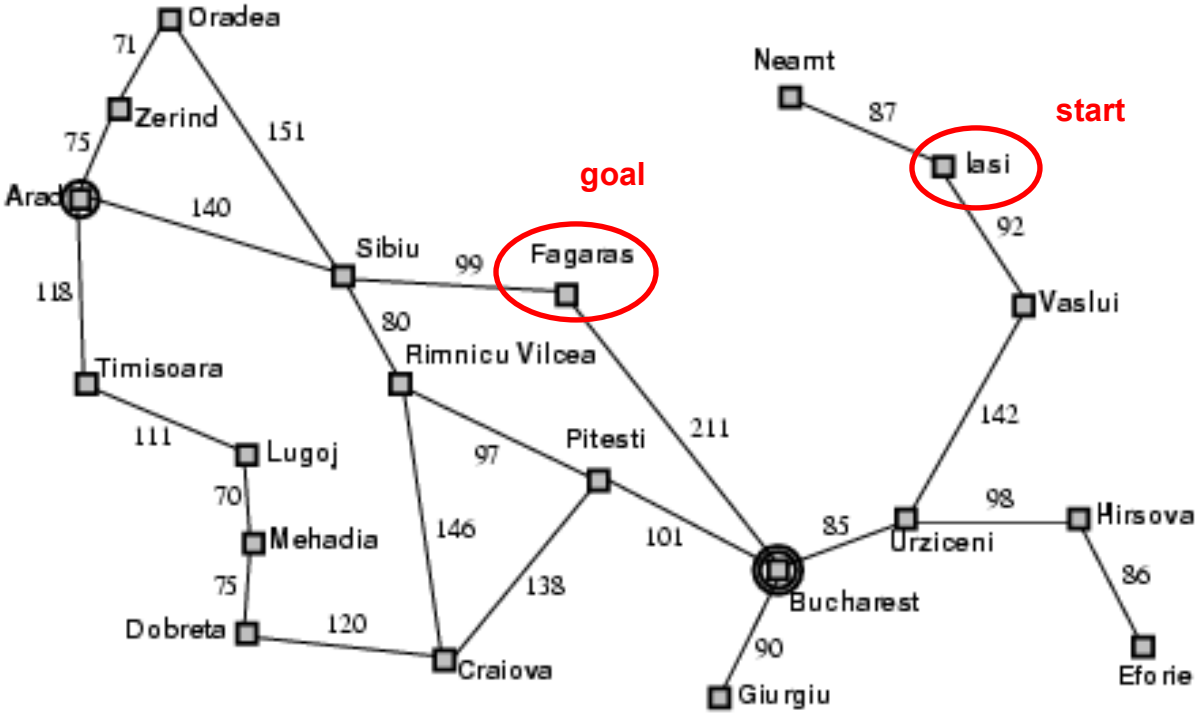


Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Properties of greedy search

Complete?

No – can get stuck in loops.



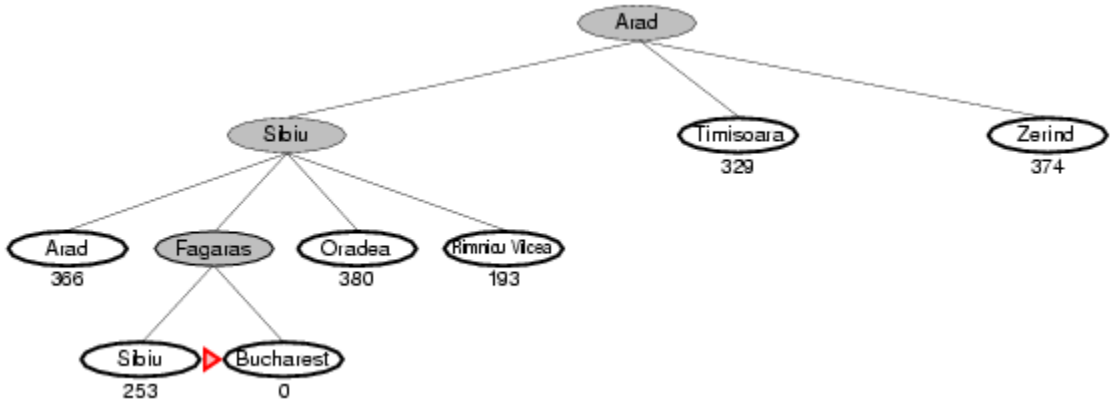
Properties of greedy best-first search

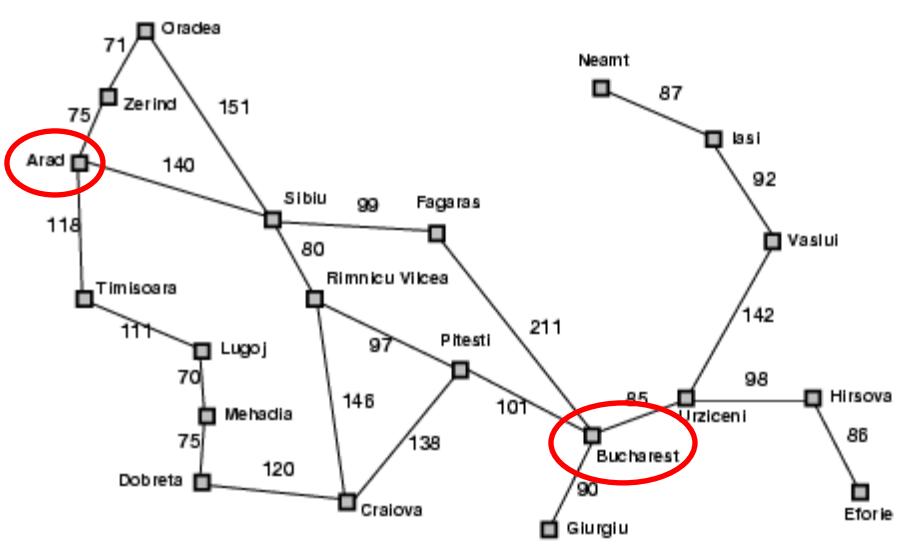
Complete?

No – can get stuck in loops

Optimal?

No





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	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Properties of greedy best-first search

Complete?

No – can get stuck in loops

Optimal?

No

Time cost?

Worst case: $O(b^m)$ where b is the branching factor and m is the maximum path length

Space cost?

Worst case: $O(b^m)$

How can we fix the greedy search?

In the Romania traveling problem, greedy search only considers the **remaining distance**

Idea: Keep track of the distance already traveled in addition to the distance remaining.

A* search

Idea: Use **evaluation function** $f(n)$ to estimate the path cost from *start node* to goal and expand the node with the lowest $f(n)$ value

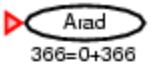
$f(n)$ is to estimate path cost from start node to goal:

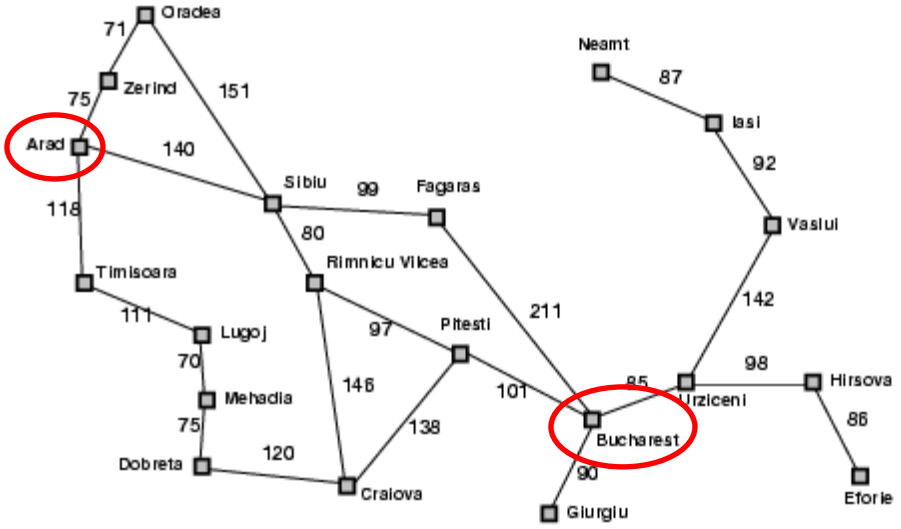
$$f(n) = g(n) + h(n)$$

$g(n)$: cost from start node to node n

$h(n)$: estimated cost from node n to goal (heuristic function)

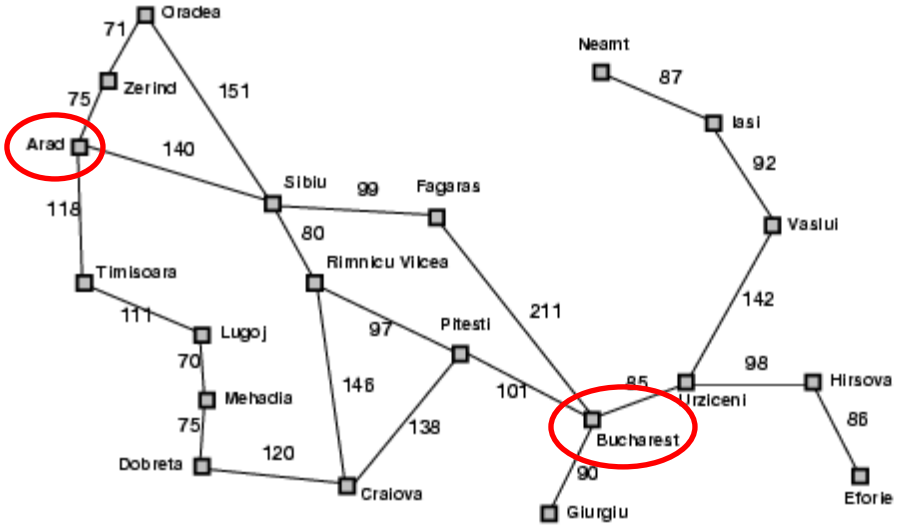
A* search example (distance already traveled + distance remaining)





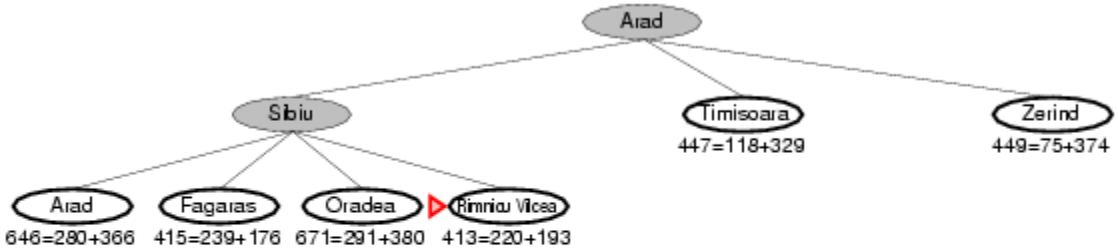
Straight–line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

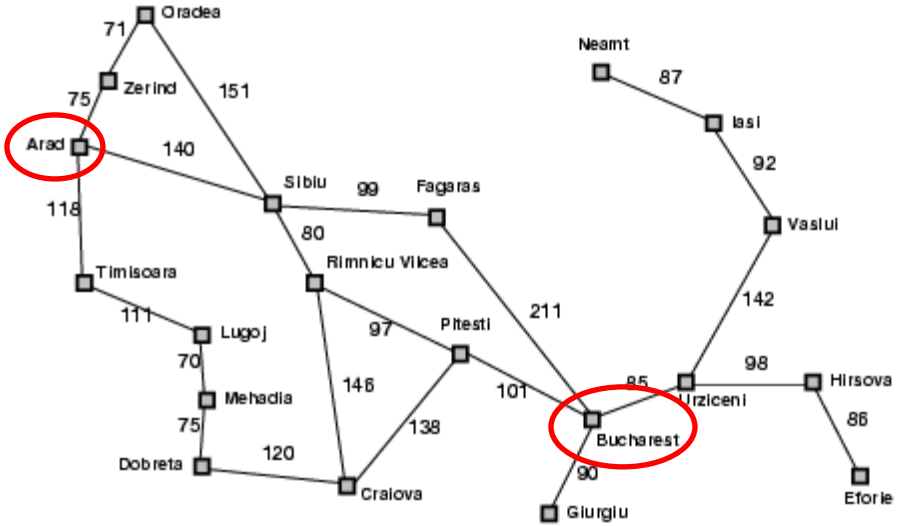
A* search example



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	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

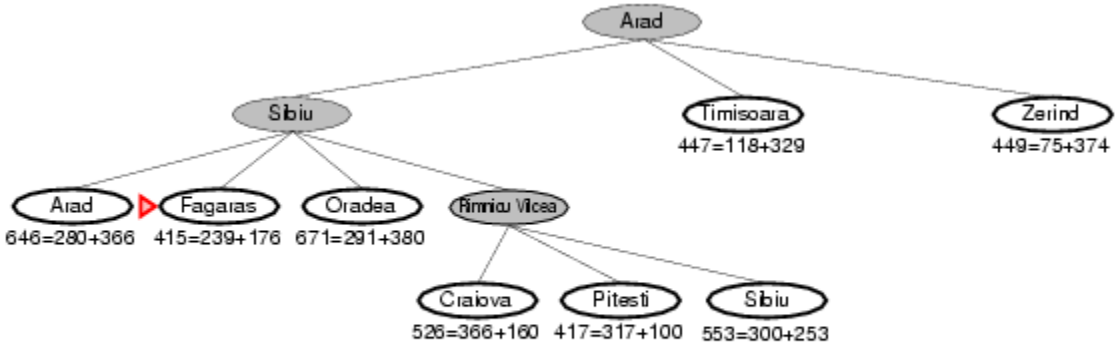
A* search example

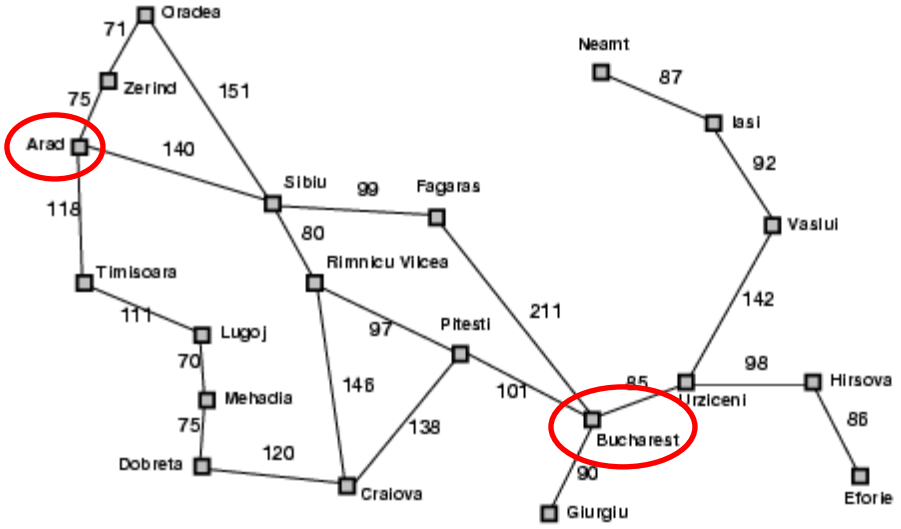




Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

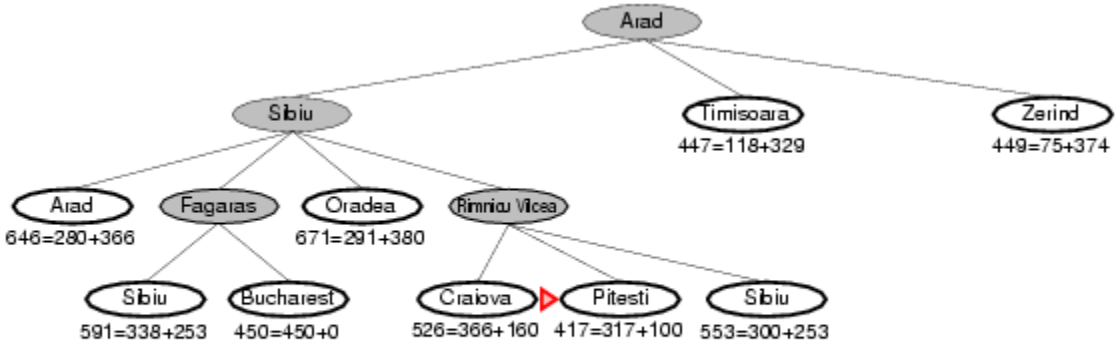
A* search example

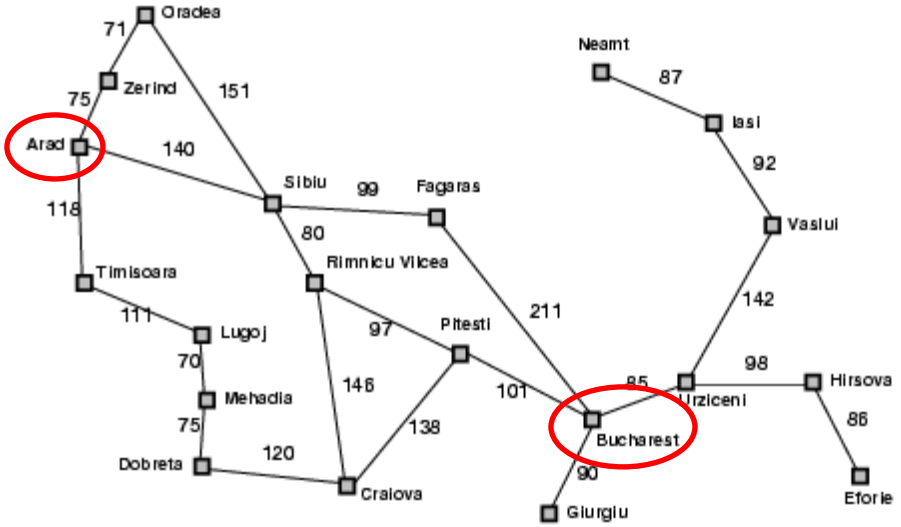




Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

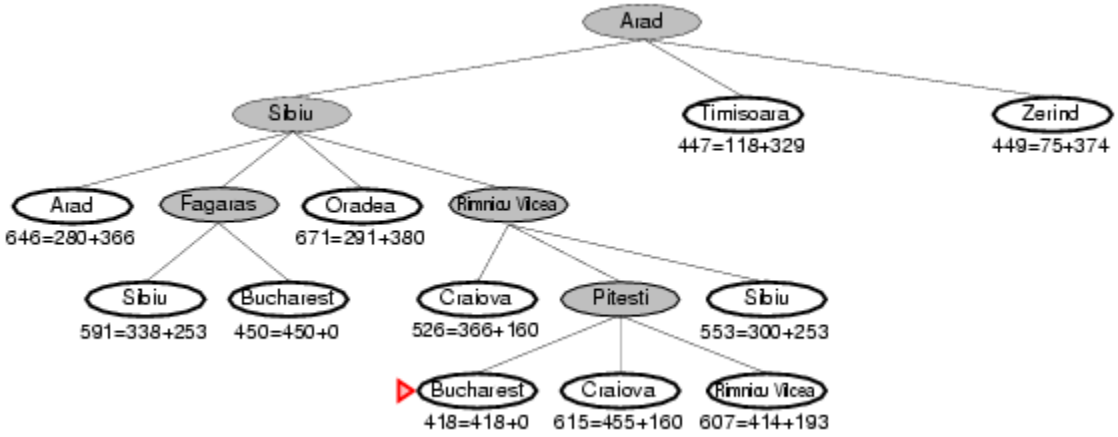
A* search example

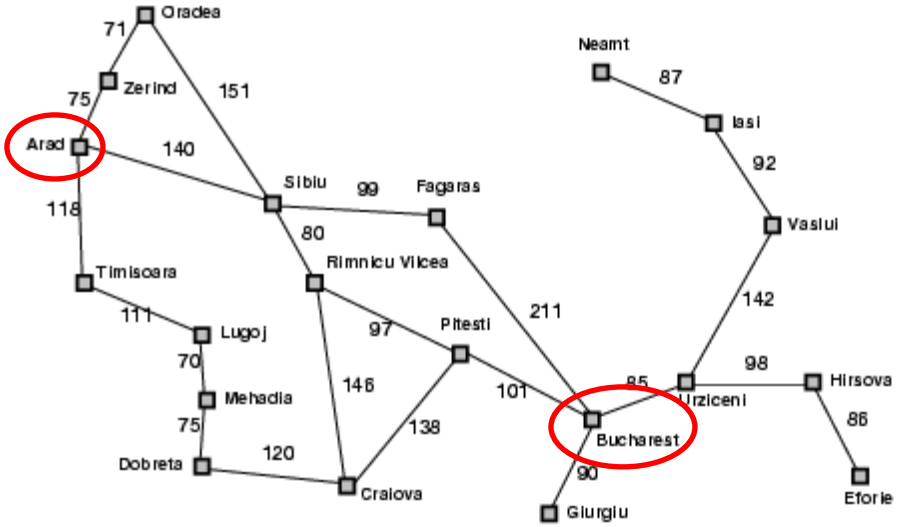




Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

A* search example

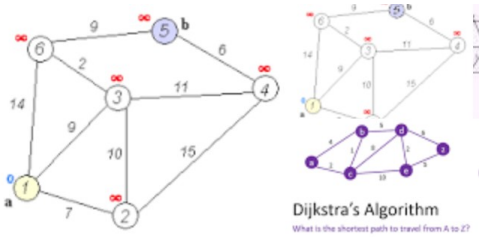




Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

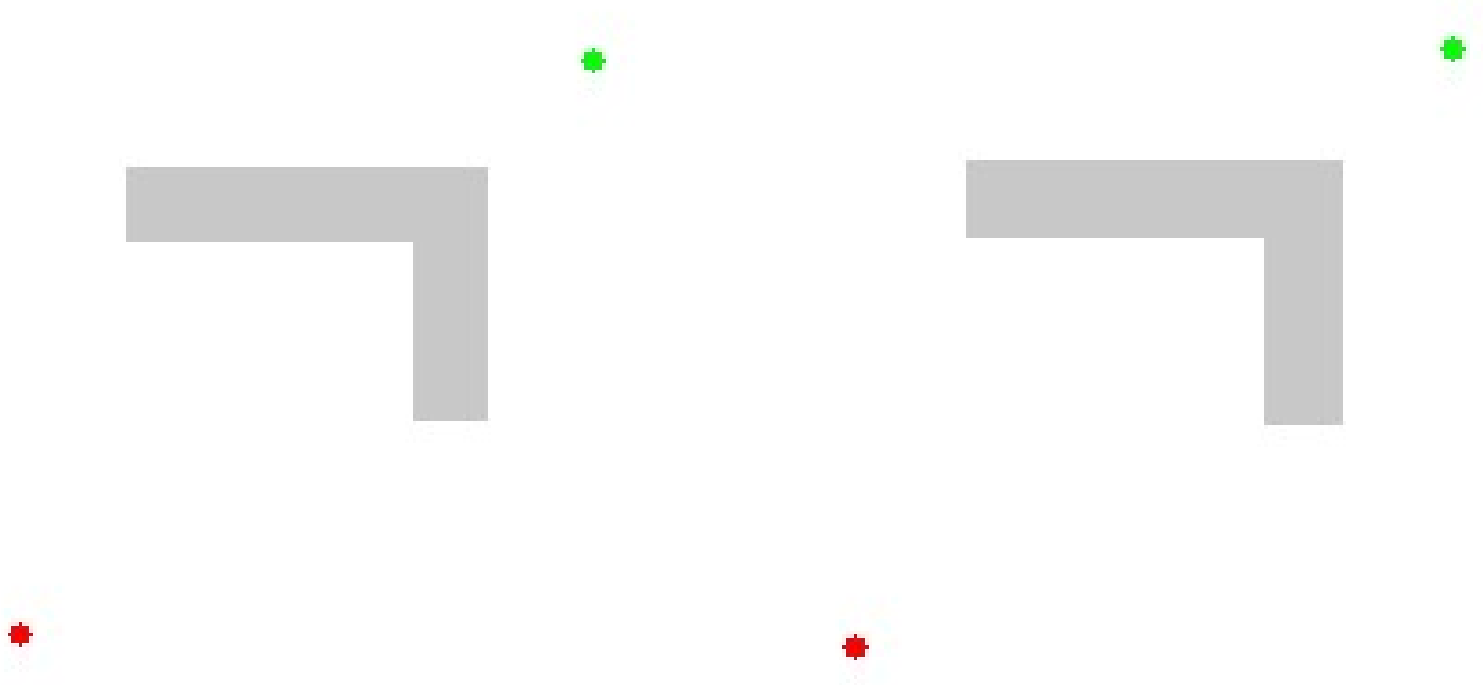
Dijkstra's vs Greedy vs A*

- Dijkstra's algorithm order all the nodes on their **g(n)** values (distance already traveled).
- Greedy algorithm order all the nodes on their **h(n)** values (distance remaining).
- A* balances Dijkstra's algorithm with greedy search by ordering all the nodes using **f(n) = g(n) + h(n),** which is the estimated path cost from **start node** through **n *to goal***.



Dijkstra's algorithm

Dijkstra's vs. A* search



Source: Wikipedia

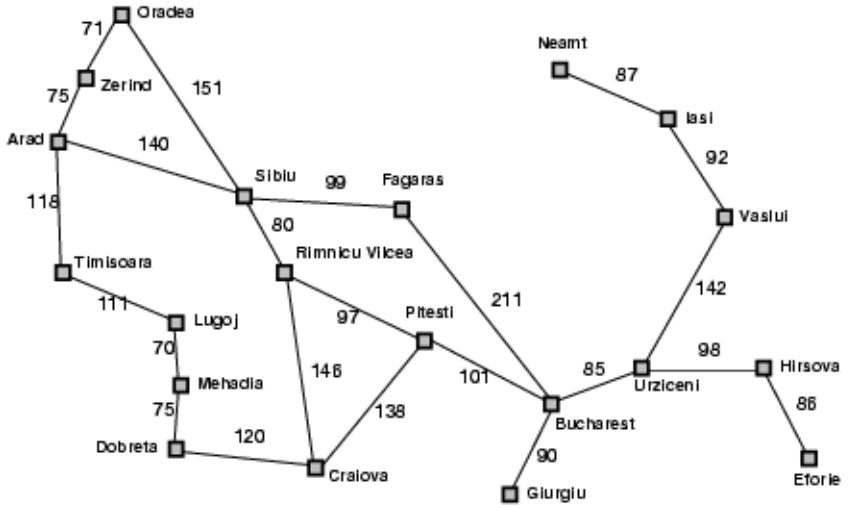
Is A* optimal?

- Is A* is guaranteed to provide the shortest path?

- **Theorem:** If $h(n)$ is admissible, then A* is optimal

Admissible heuristic

- A heuristic $h(n)$ is admissible if for every node n, $h(n) \leq h^*(n)$, where $h(n)$ is the estimated cost from n to goal while $h^*(n)$ is the true cost from n to goal.
- Is the distance heuristic we used before in Romanian traveling problem is admissible?



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	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

- Yes. Because straight line distance never overestimates the

How to tell which heuristic is better?

If h_1 and h_2 are both admissible heuristics and $h_2(n) \geq h_1(n)$ for all n, then h_2 dominates h_1

Using $h_2(n)$ will lead to expanding fewer nodes than $h_1(n)$.

Properties of A*

Complete?

Yes – Will always find a solution if one exists

Optimal?

- Yes If $h(n)$ is admissible

Time/space cost?

- A* search expands every node with $h(n) \leq C^* - g(n)$
  - C^* is the path cost of the optimal solution
  - $g(n)$: cost from start node to node n
  - *A higher (dominating)* $h(n)$ will have a higher pruning power

Search strategies

Algorithm	Complete?	Optimal?	Time complexity	Space complexity
Greedy	No	No	Worst case: $O(b^m)$ where b is branching factor and m is maximum path length	
A*	Yes	Yes (if heuristic is admissible)	Number of nodes with $h(n) \leq C^* - g(n)$ $h(n)$: estimated cost from n to goal $g(n)$: cost from start node to n C^* is the path cost of the optimal solution	