Mansoura University
Faculty of Computers and Information
Department of Computer Science
First Semester: 2020-2021

[CS324P] Artificial Intelligence - 1 : Solving Problems By Searching

Grade: Third Year (Computer Science)

Ass. Prof. Taher Hamza

Dr. Sara El-Metwally

Faculty of Computers and Information,

Mansoura University,

Egypt.

Contents

Heuristics

Uninformed vs. Informed

Uninformed

Can only generate successors and distinguish goals from non-goals

Informed

Strategies that know whether one non-goal is more promising than another

Note

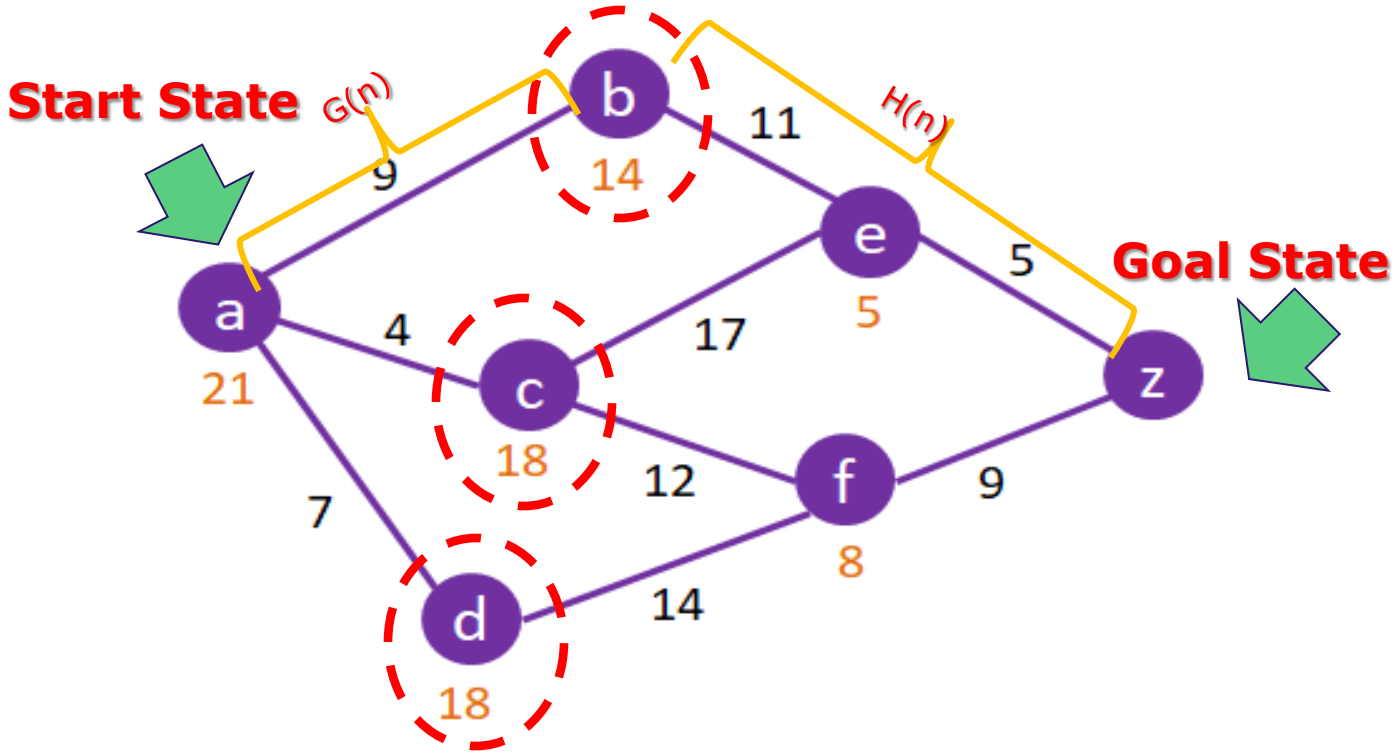


Image credit: https://www.101computing.net/a-star-search-algorithm/

Heuristics

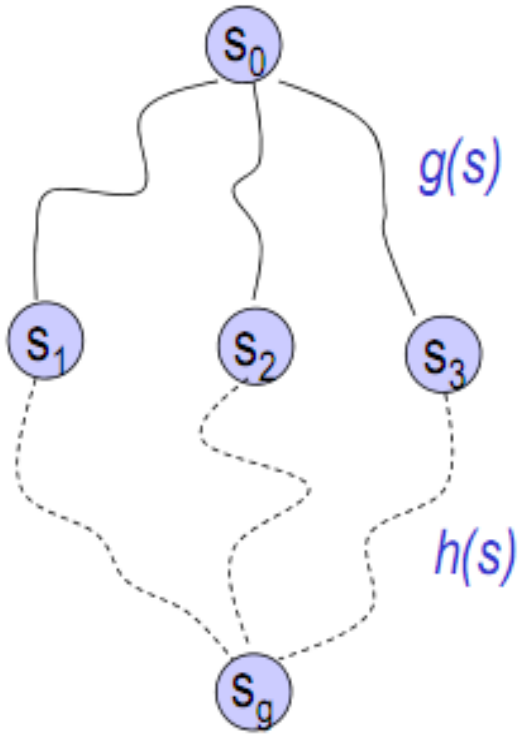
- Extra information used by the searching algorithm beside problem definition
- Heuristic is an estimated cost to goal node
- Recall, heuristic is not actual cost it is just an expectation (**optimistic**)

Heuristics

g(s): this is a function that measures the "cost" it incurred from the initial node s_o to the current node s.

h(s): this is a function (or "budget") that estimates the forth-coming cost from *s* to a goal node s_g.

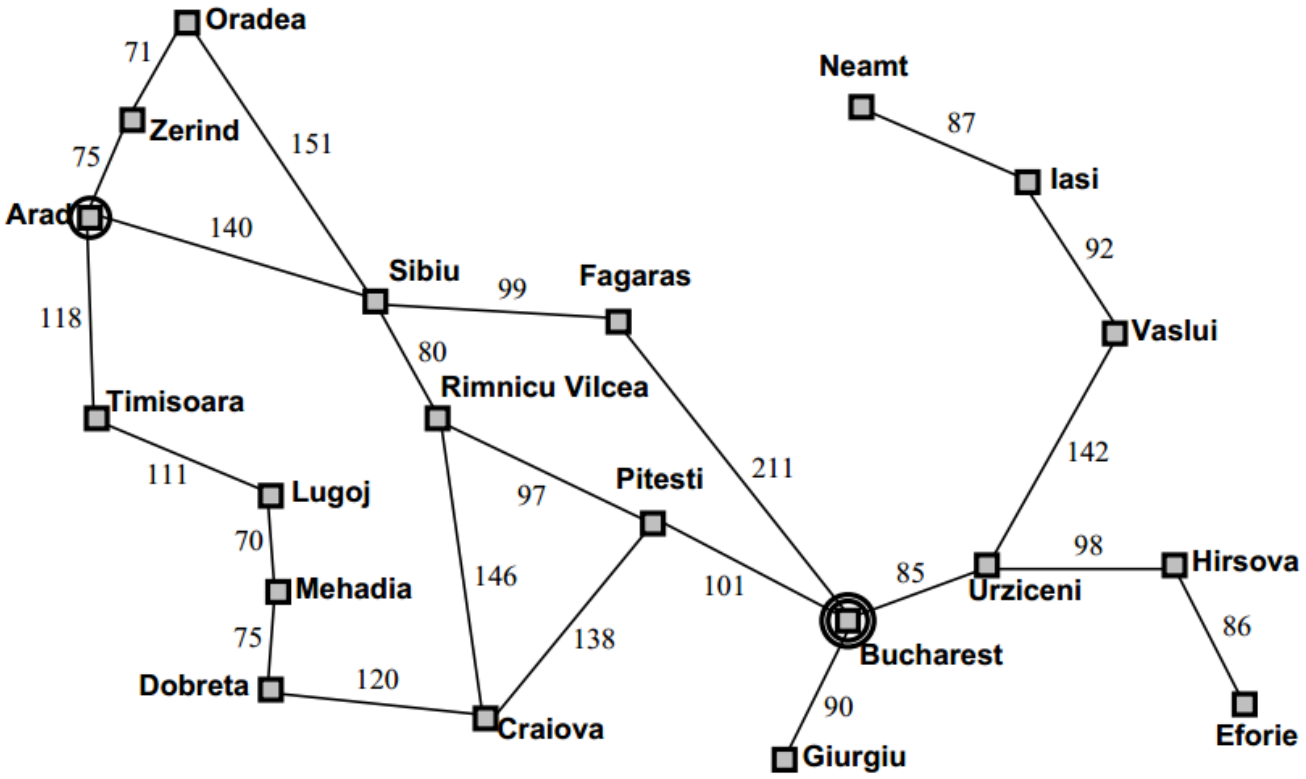
h(s) is called a ***heuristic function****.*



Heuristics, example

Route finding problems:

- h(n) = straight line distance (SLD) from node to goal

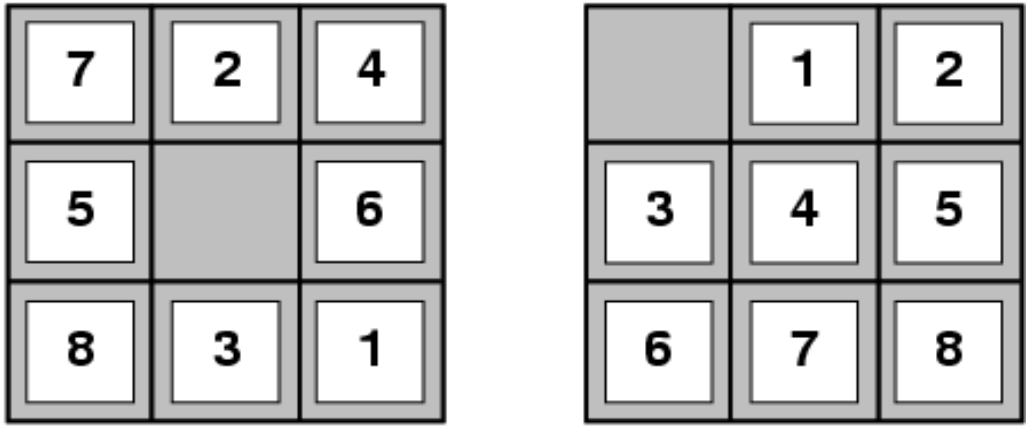


Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Heuristics, example

The 8-puzzle:

h(n) = number of misplaced tiles

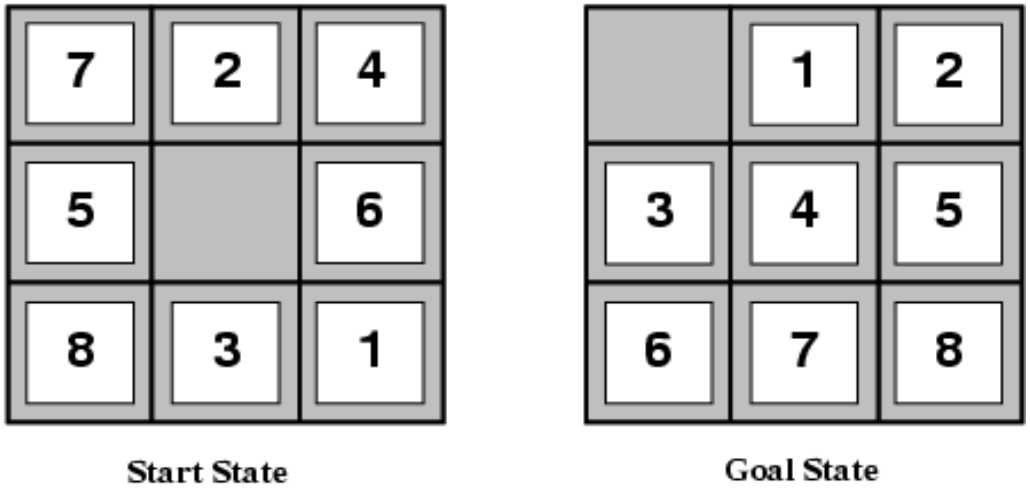


h(node) = 8

Heuristics, example

The 8-puzzle:

h(n) = total Manhattan distance (i.e., no. of squares from desired location of each tile)



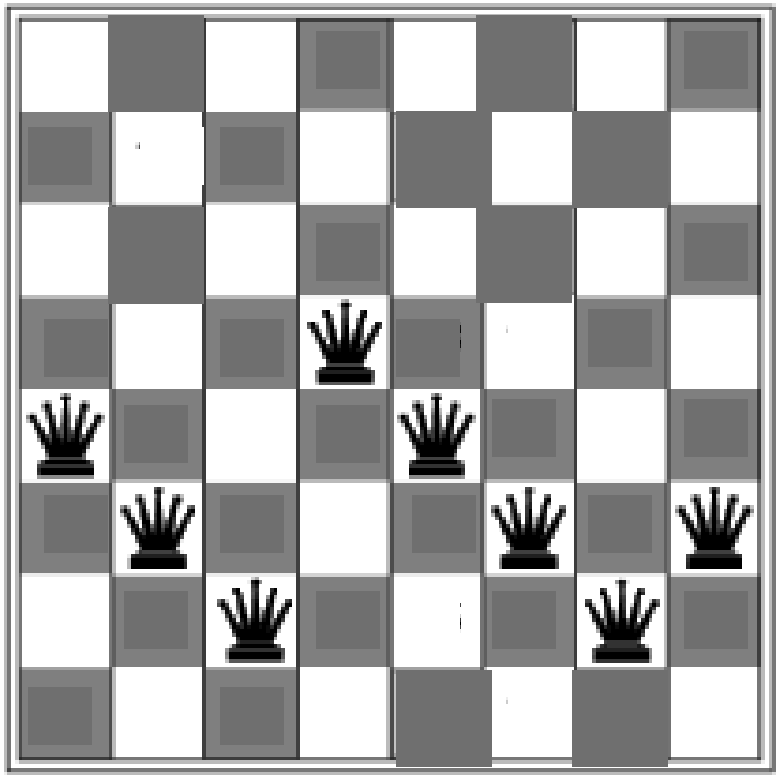
h(node) = 3+1+2+2+2+3+3+2 = 18

Heuristics, example

The 8-queens:

h(n) = number of pairs of queens that are attacking each other, either directly or indirectly

h(node) = 17



Informed search algorithms

- Use the heuristic function in order to optimize the search

- Informed search algorithms:
  1. Best-first search
     - 1.1 Greedy search
     - 1.2 A* search
  2. Local search
     - 2.1 Hill climbing algorithm
     - 2.2 Genetic algorithm

Best first search

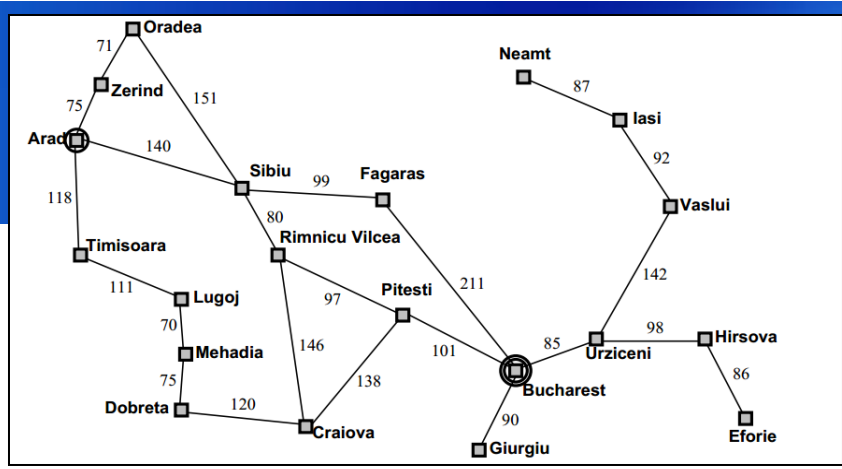
1. Greedy search

- Evaluation function f(n) = h(n)
- h(n) is the heuristic function
- Greedy best-first search expands the node that appears to be closest to goal

choose node with minimum f(n)

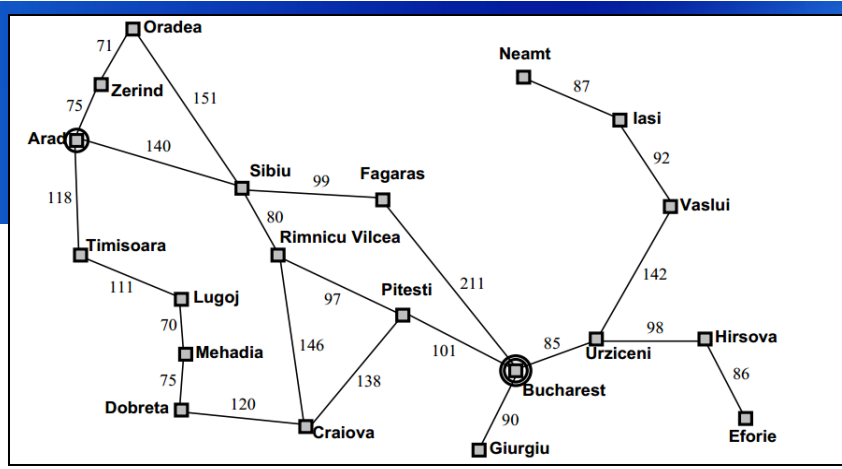
Best first search

1. Greedy search, example

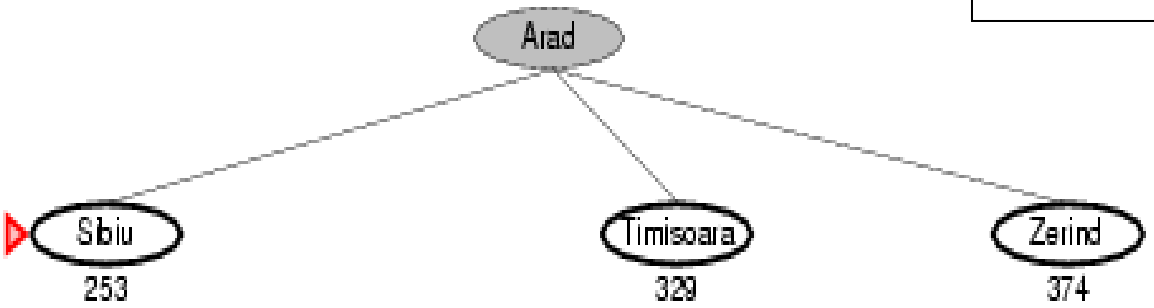


Straight–line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Best first search

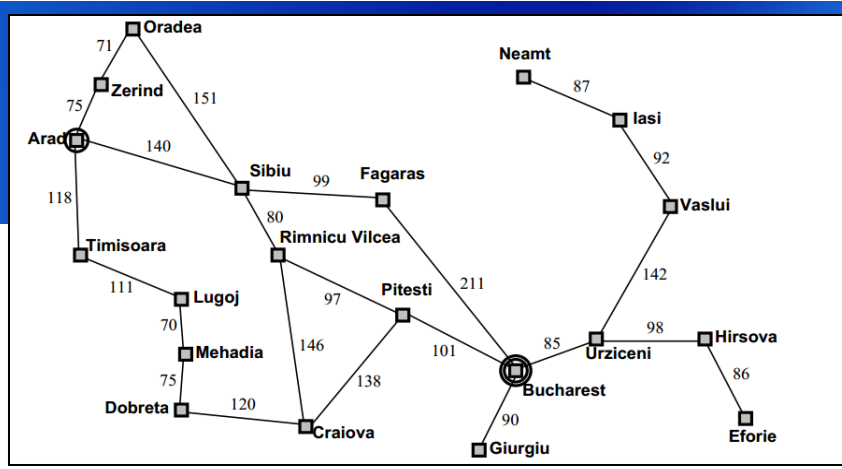


1. Greedy search, example

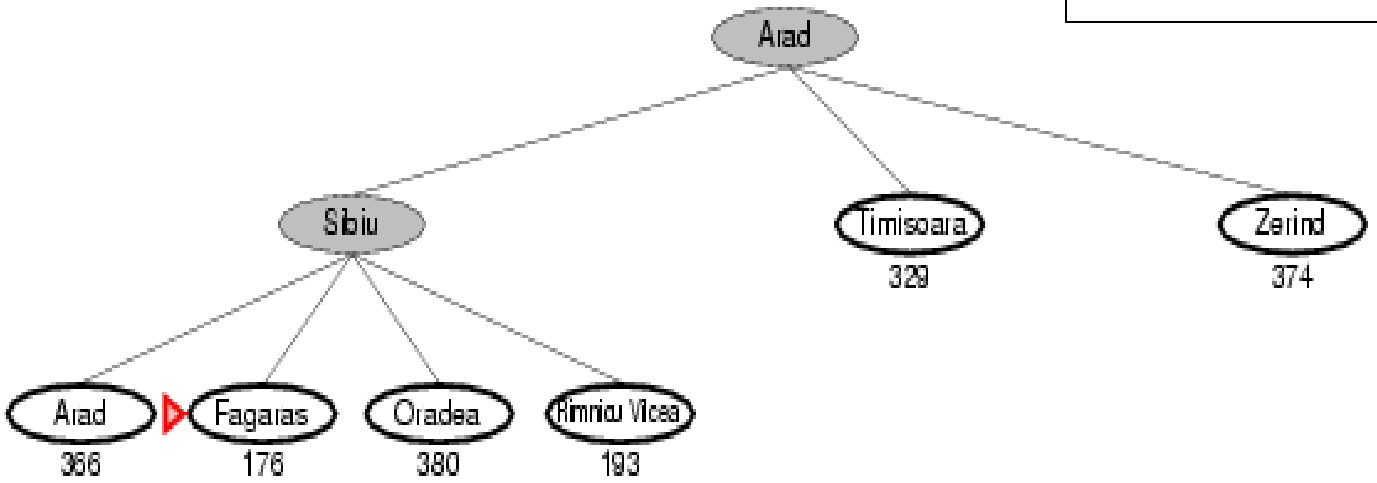


Straight–line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Best first search

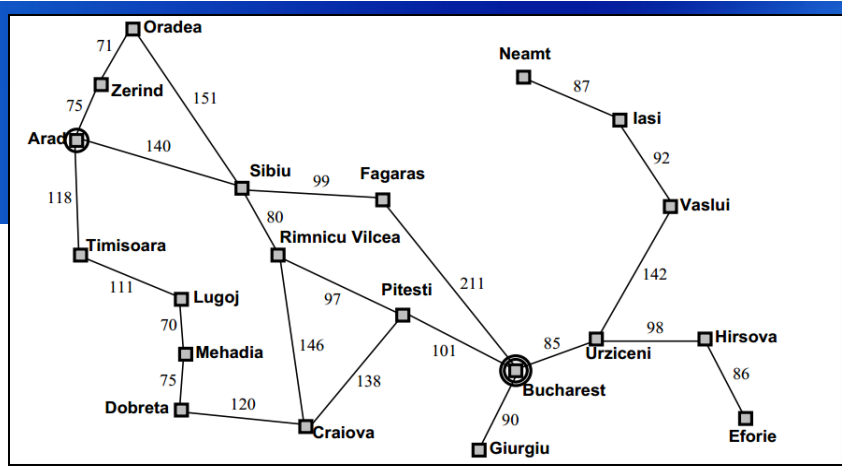


1. Greedy search, example

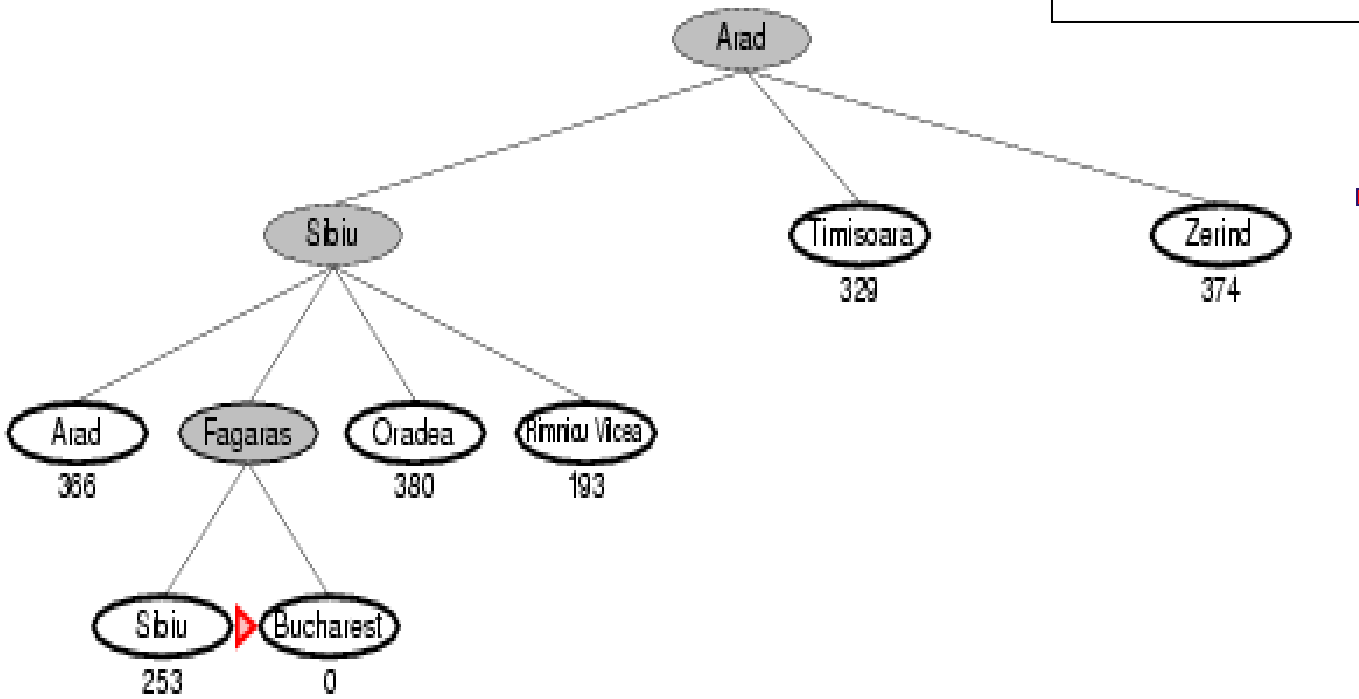


Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Best first search



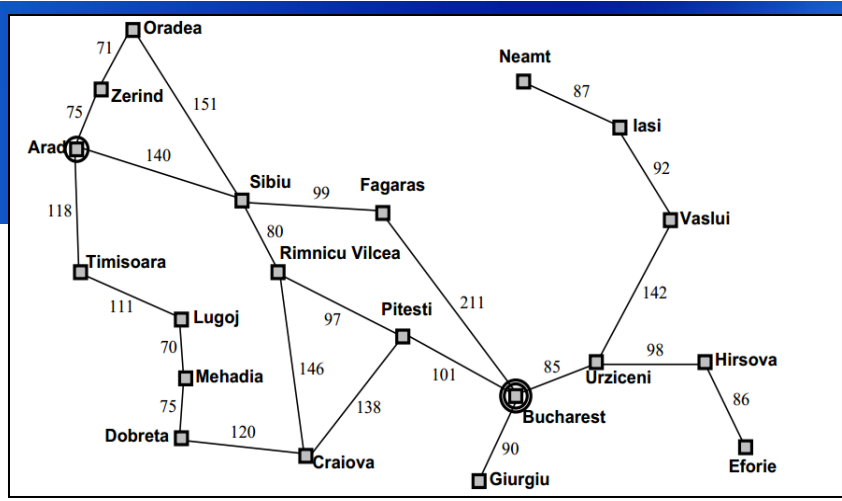
1. Greedy search, example



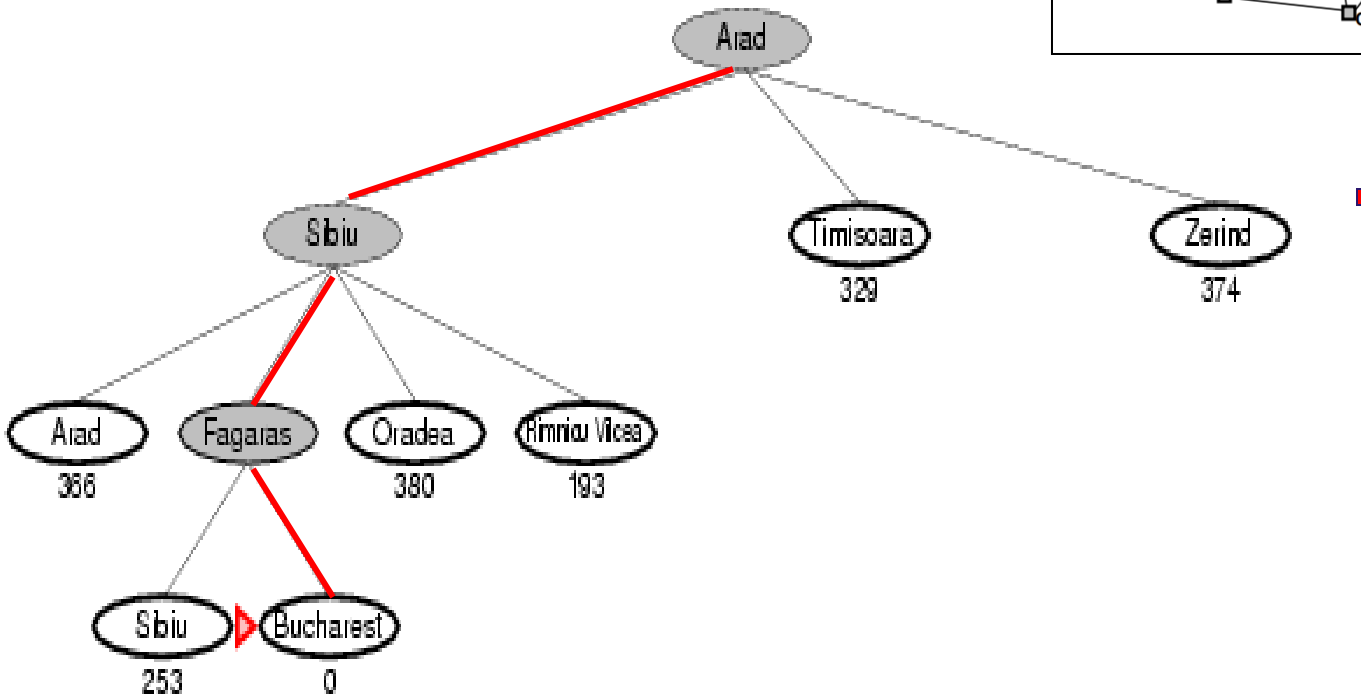
Goal is found!

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

Best first search



1. Greedy search, example

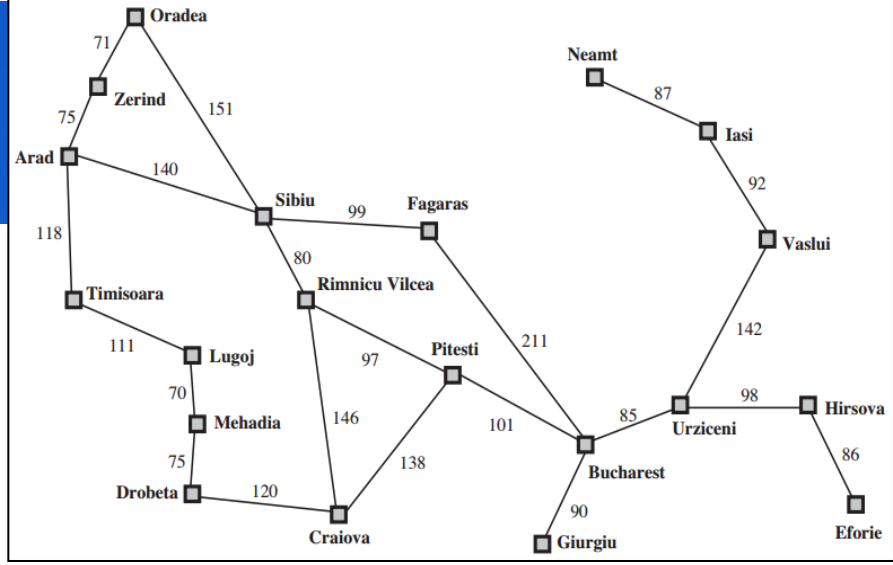


Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Total coast=140+99+211= 450

Is this the optimum solution ?

Evaluation



1. Greedy search

❖ Complete?

- Suppose we start in Iasi and the goal is Fagaras
- Greedy select Neamt for expand because it is closest to Fagaras
- But it is dead end, it only generate Iasi again
- It will stuck in the loop: Iasi → Neamt → Iasi → Neamt,……

Evaluation

1. Greedy search

- Complete? No (can stuck on loops)
- Time? Exponential (good heuristic can give dramatic improvement)
- Space? keeps all nodes in memory **(look at this)**
- Optimal? No

Best first search

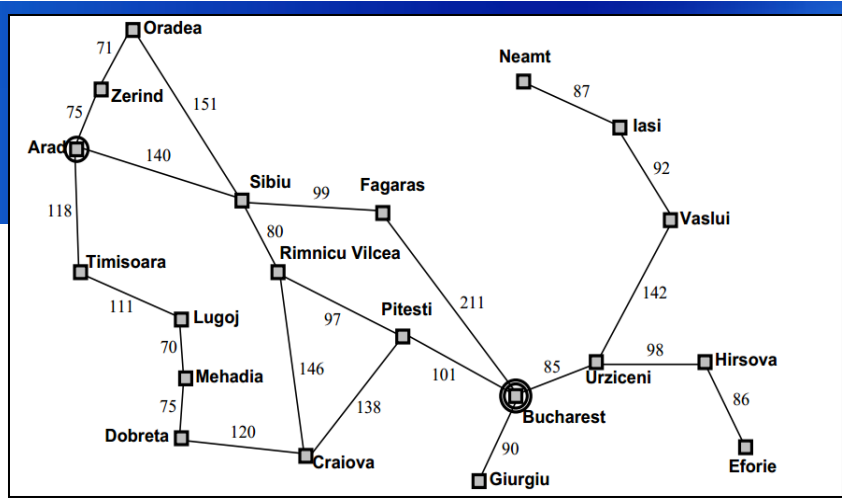
2. A* search

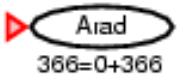
- Avoid expanding paths that are already expensive
- Evaluation function f(n) = g(n)+h(n)

  - g(n) = cost so far to reach n (actual)
  - h(n) = expected cost from n to goal (estimated)

Best first search

2. A*search, example

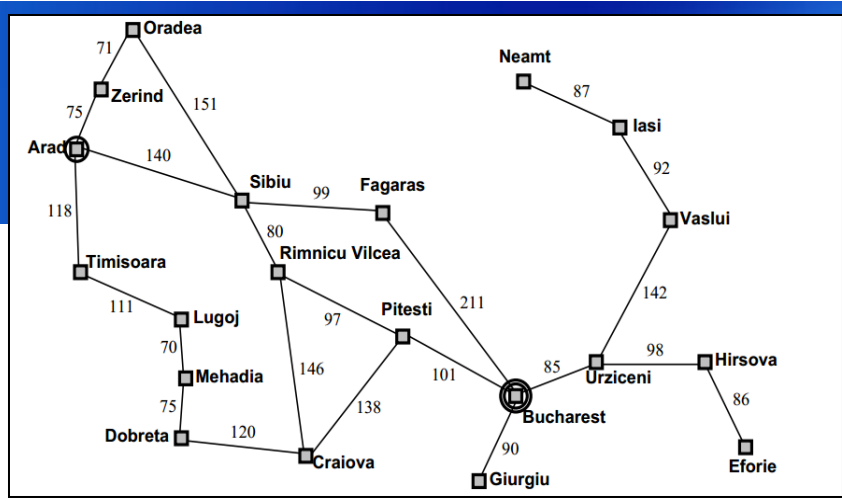




Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Best first search

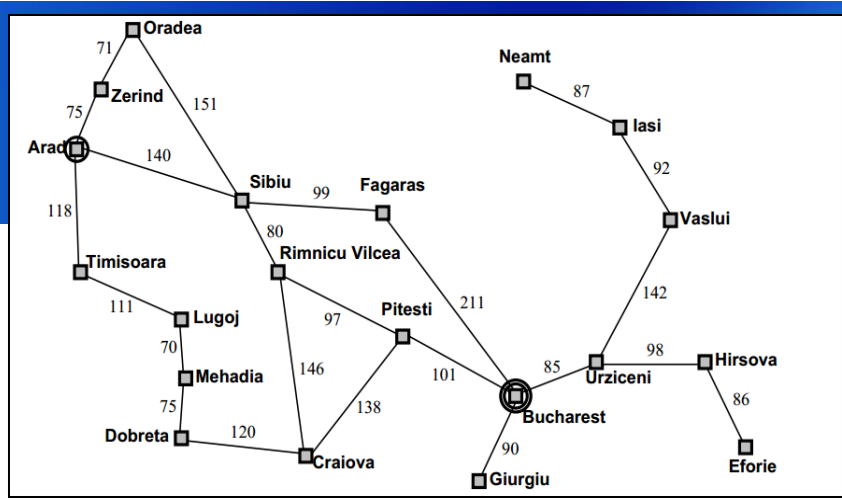
2. A*search, example



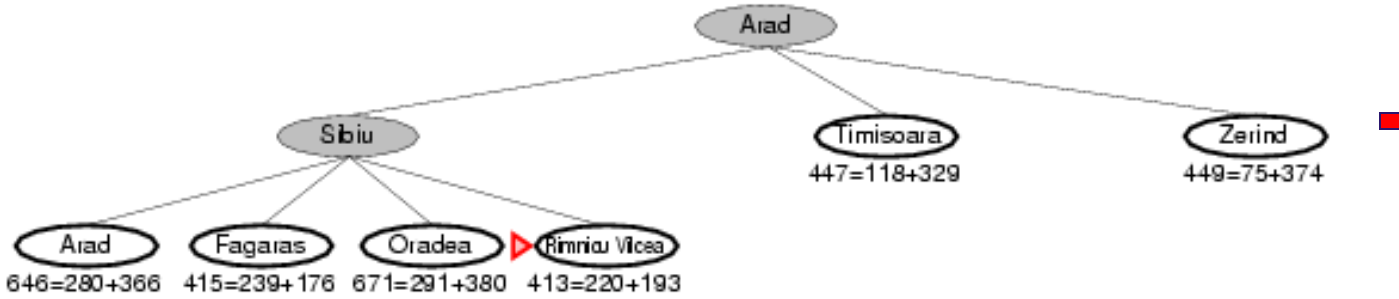
Straight–line distance to Bucharest

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

Best first search

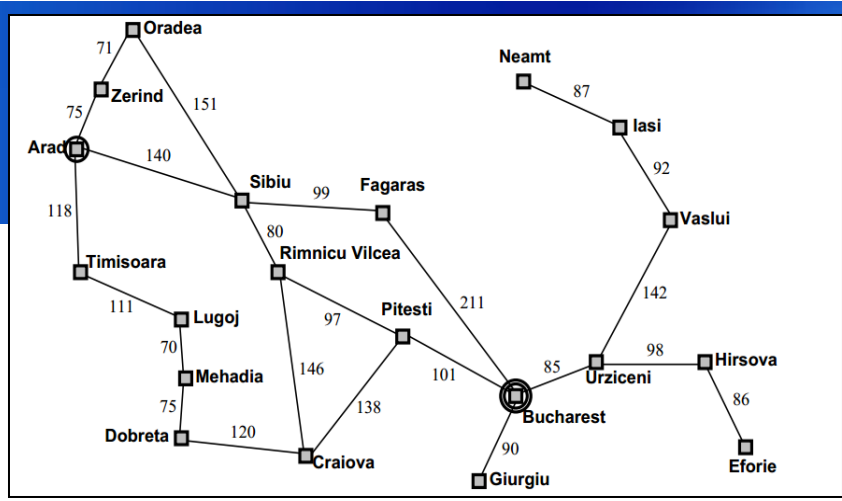


2. A*search, example

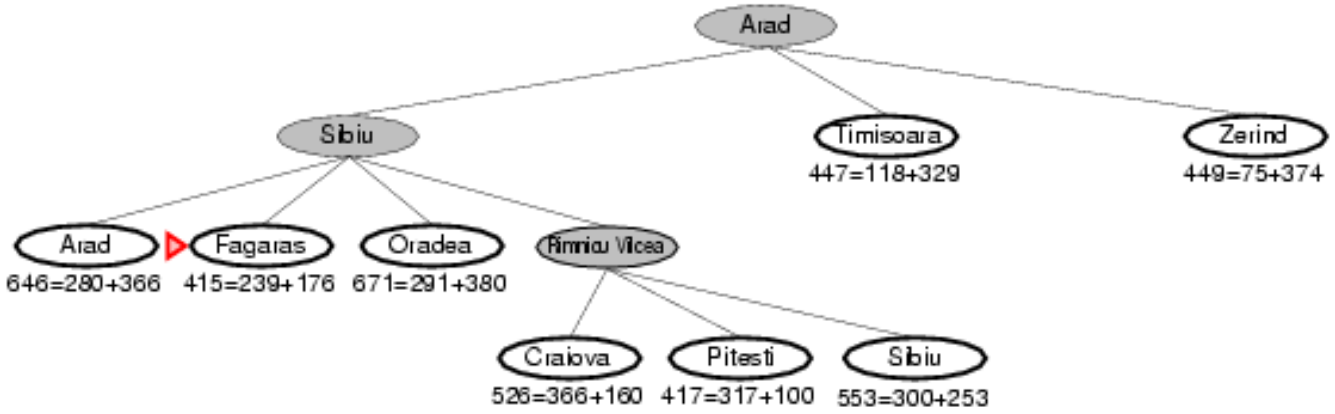


Straight–line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Best first search



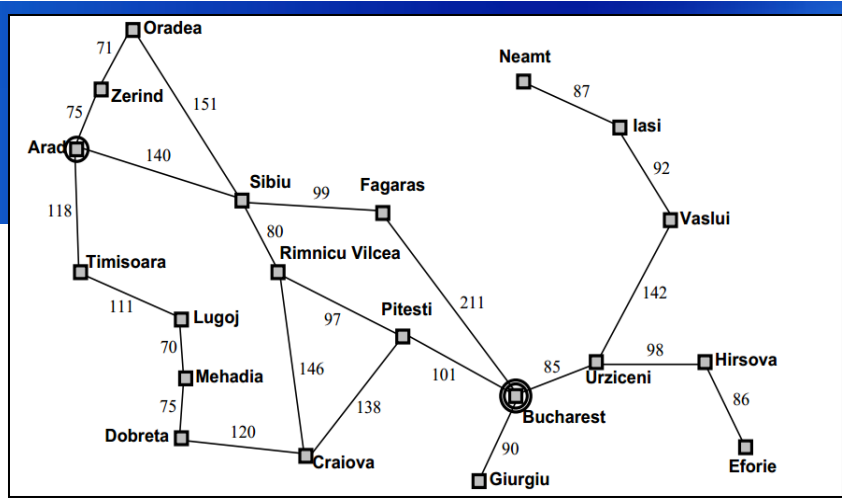
2. A*search, example

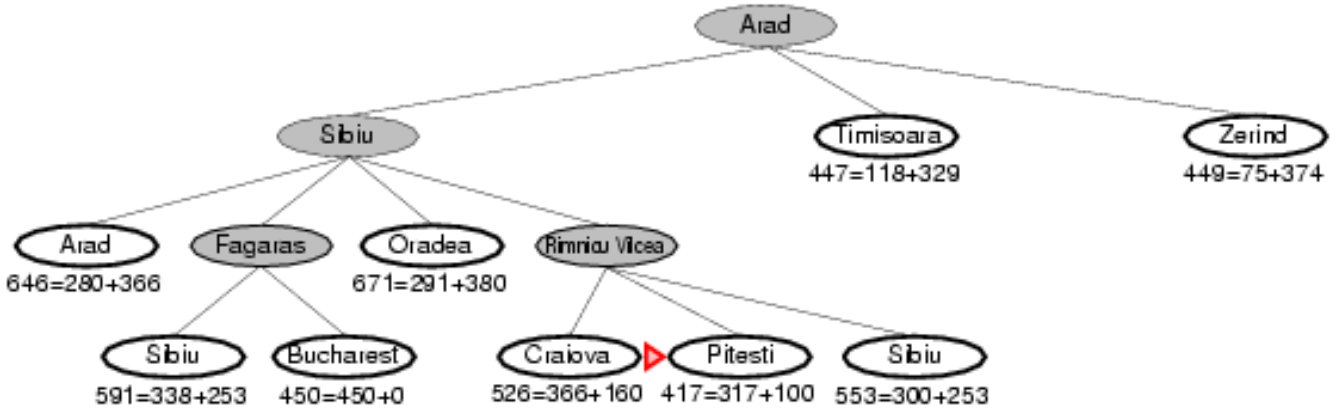


Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Best first search

2. A*search, example

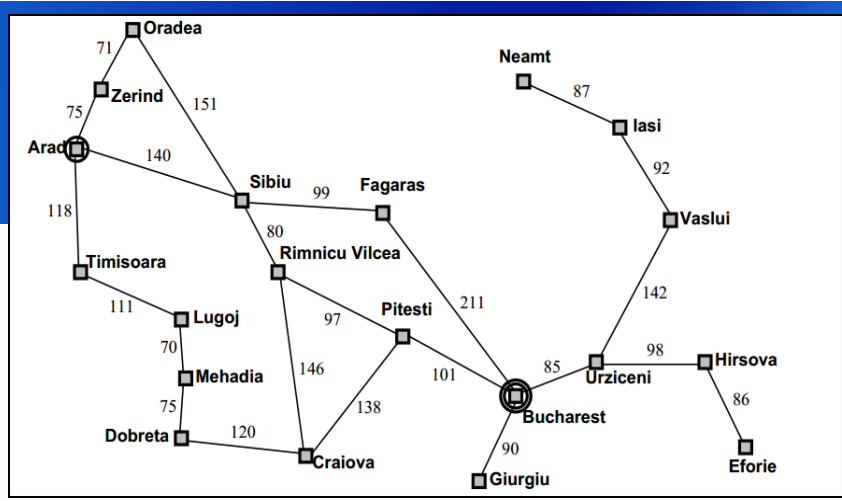


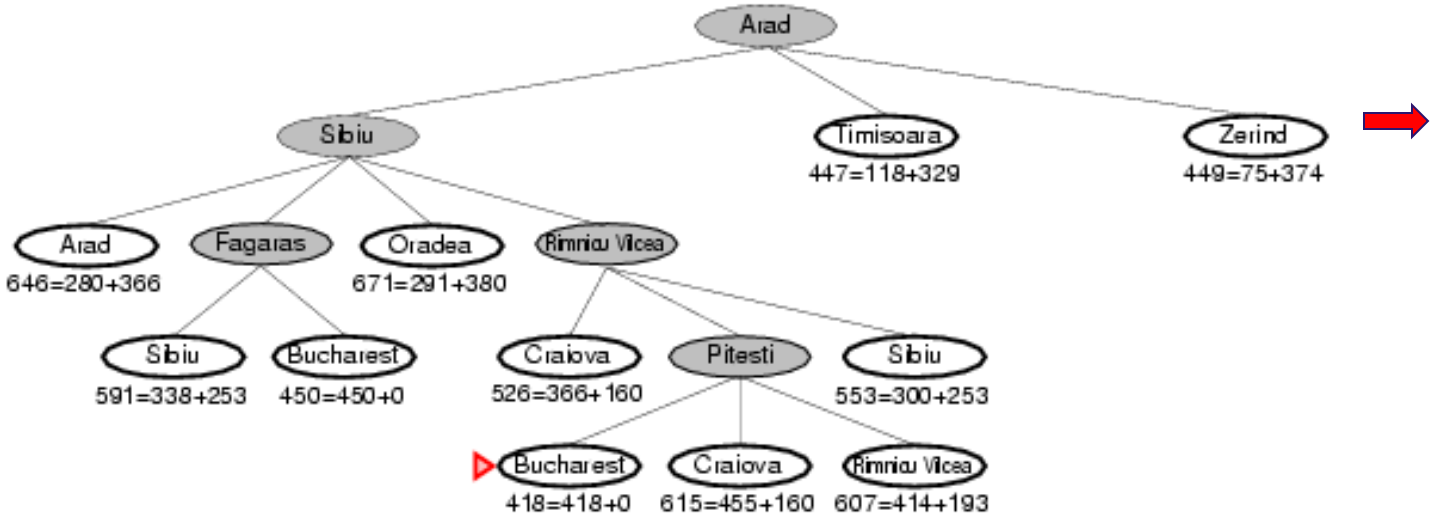


Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Best first search

2. A*search, example

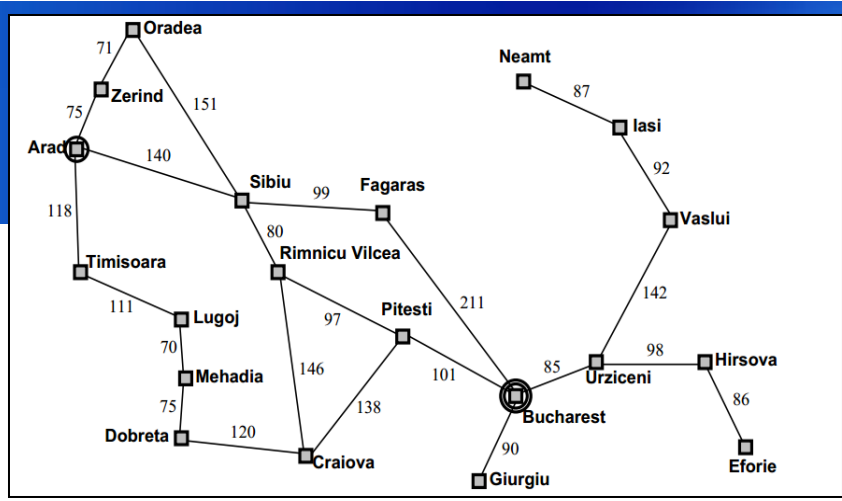




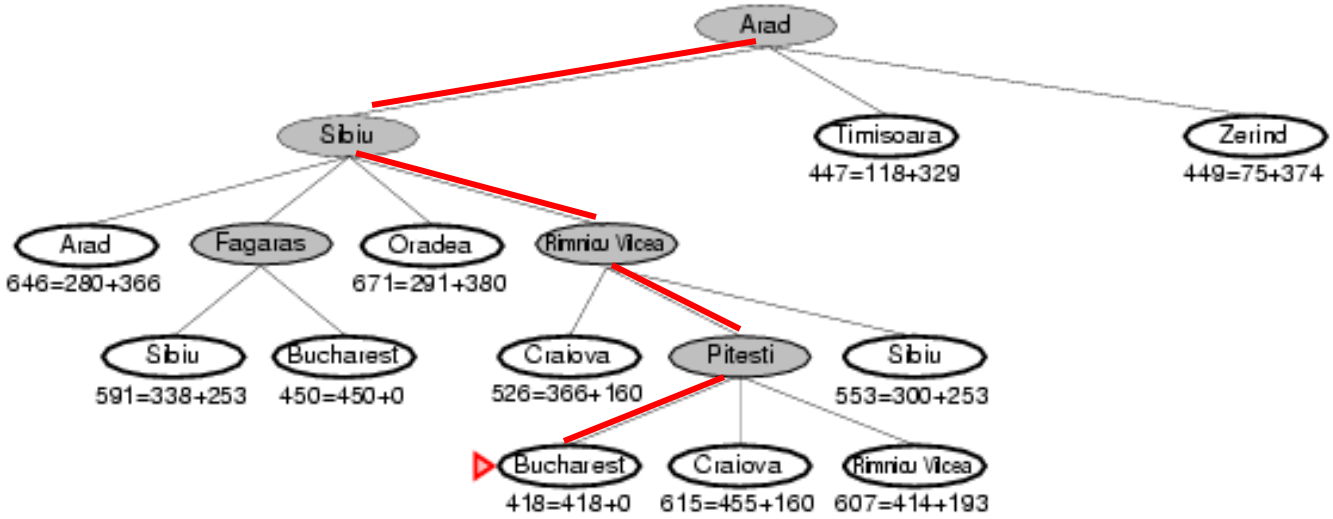
Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Goal is found!

Best first search



2. A*search, example



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

Total coast =418

Is this the optimum solution ? yes

Evaluation

2. A* search

- Complete? yes
- Time? Exponential
- Space? keeps all nodes in memory (look at this)
- Optimal? yes

Evaluation

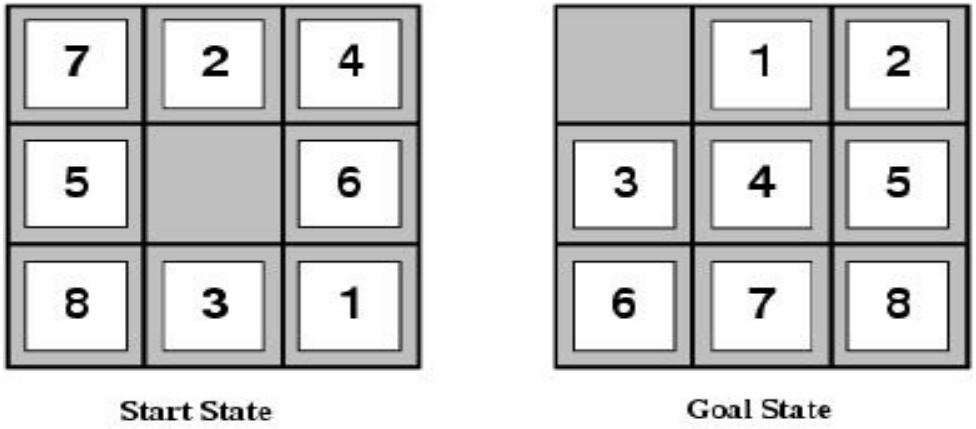
- Heuristic function should be admissible in order to find optimum solutions

- A heuristic h(n) is admissible if for every node n, **h(n) ≤ h*(n)** ,where h*(n) is the true (actual) cost to reach the goal state from n

- An admissible heuristic **never overestimates** the cost to reach the goal, i.e., it is optimistic

Evaluation

Dominance



- **h1** = the number of misplaced tiles **=8**
- **h2** = manhattan distance**= 18**

iff **h2(n) >= h1(n)** for all n **(both admissible)**
then h2 dominates h1,

- **h2 is better for search than h1**

Evaluation

Dominance

The following table gives the search cost of A* with the two heuristics, averaged over random 8-puzzles, for various solution lengths

Length	$A^*(h_1)$	$A^*(h_2)$
16	1301	211
18	3056	363
20	7276	676
22	18094	1219
24	39135	1641


Thank You !