

# Informed search algorithms

## Chapter 4

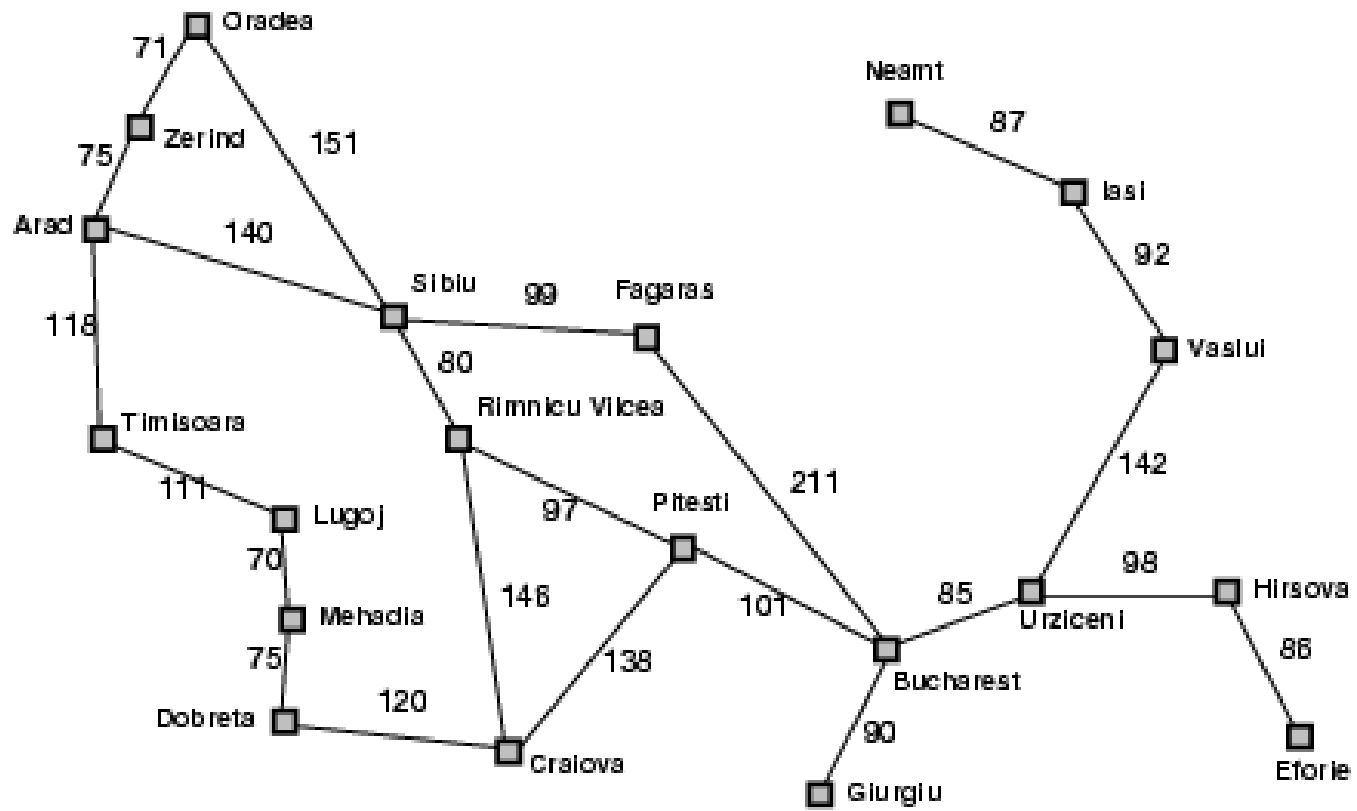
# Outline

- Best-first search
- Greedy best-first search
- A\* search
- Heuristics
- Local search algorithms
- Hill-climbing search
- Simulated annealing search
- Local beam search
- Genetic algorithms

# Best-first search

- Idea: use an **evaluation function**  $f(n)$  for each node
  - estimate of "desirability"
  - Expand most desirable unexpanded node
- Implementation:  
Order the nodes in fringe in decreasing order of desirability
- Special cases:
  - greedy best-first search
  - A\* search
  -
- - 
  -

# Romania with step costs in km



Straight-line distance  
to Bucharest

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

# Greedy best-first search

- Evaluation function  $f(n) = h(n)$  (**h**euristic)
- = estimate of cost from  $n$  to *goal*
- e.g.,  $h_{SLD}(n)$  = straight-line distance from  $n$  to Bucharest
- Greedy best-first search expands the node that **appears** to be closest to goal
- 
- 
-

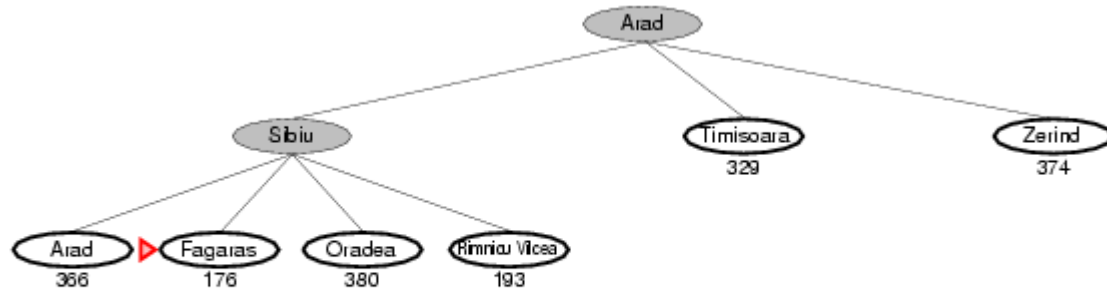
# Greedy best-first search example



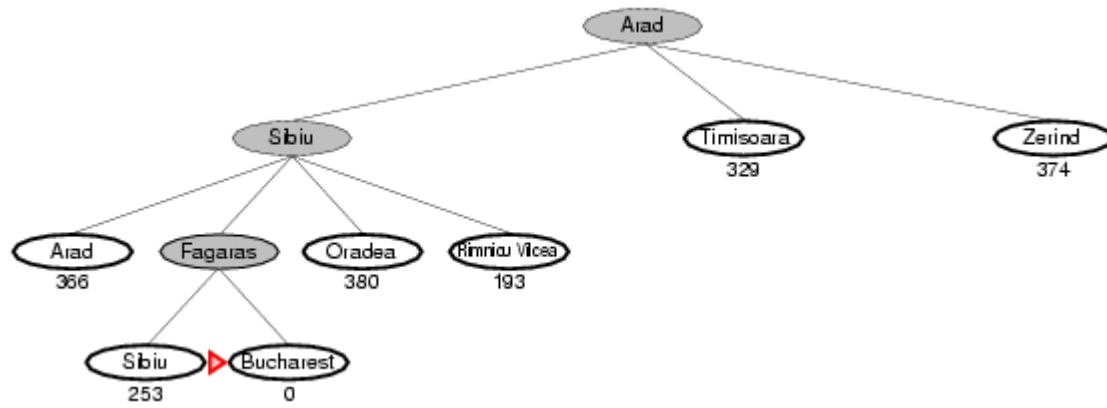
# Greedy best-first search example



# Greedy best-first search example



# Greedy best-first search example



# Properties of greedy best-first search

- Complete? No – can get stuck in loops, e.g., lasi → Neamt → lasi → Neamt →
- Time?  $O(b^m)$ , but a good heuristic can give dramatic improvement
- Space?  $O(b^m)$  -- keeps all nodes in memory
- Optimal? No
- $b$  is the branching factor of the graph
- $m$  is the maximum depth of the search space
-