# Informed (Heuristic) Search Strategies

- Evaluation function $f(n)$
- Heuristic function $h(n)$

# Informed (Heuristic) Search Strategies

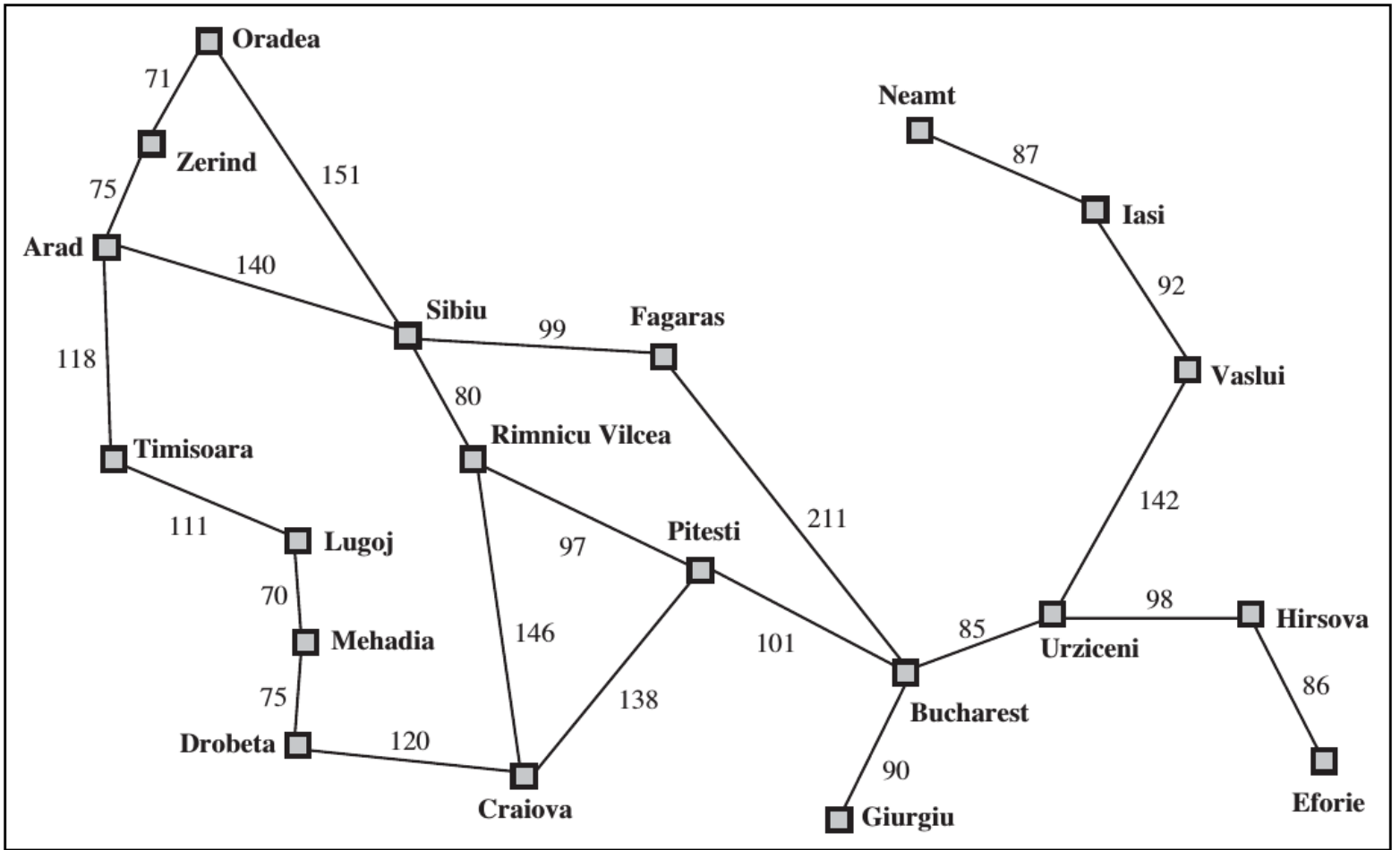



**Figure 3.2** A simplified road map of part of Romania.

# Straight Line Distances

| | | | |
|---|---|---|---|
| **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 |

**Figure 3.22** Values of $h_{SLD}$—straight-line distances to Bucharest.

# Greedy Best-first Search

- Evaluation function $f(n) = h(n)$

# Greedy Best-first Search

- Evaluation function $f(n) = h(n)$

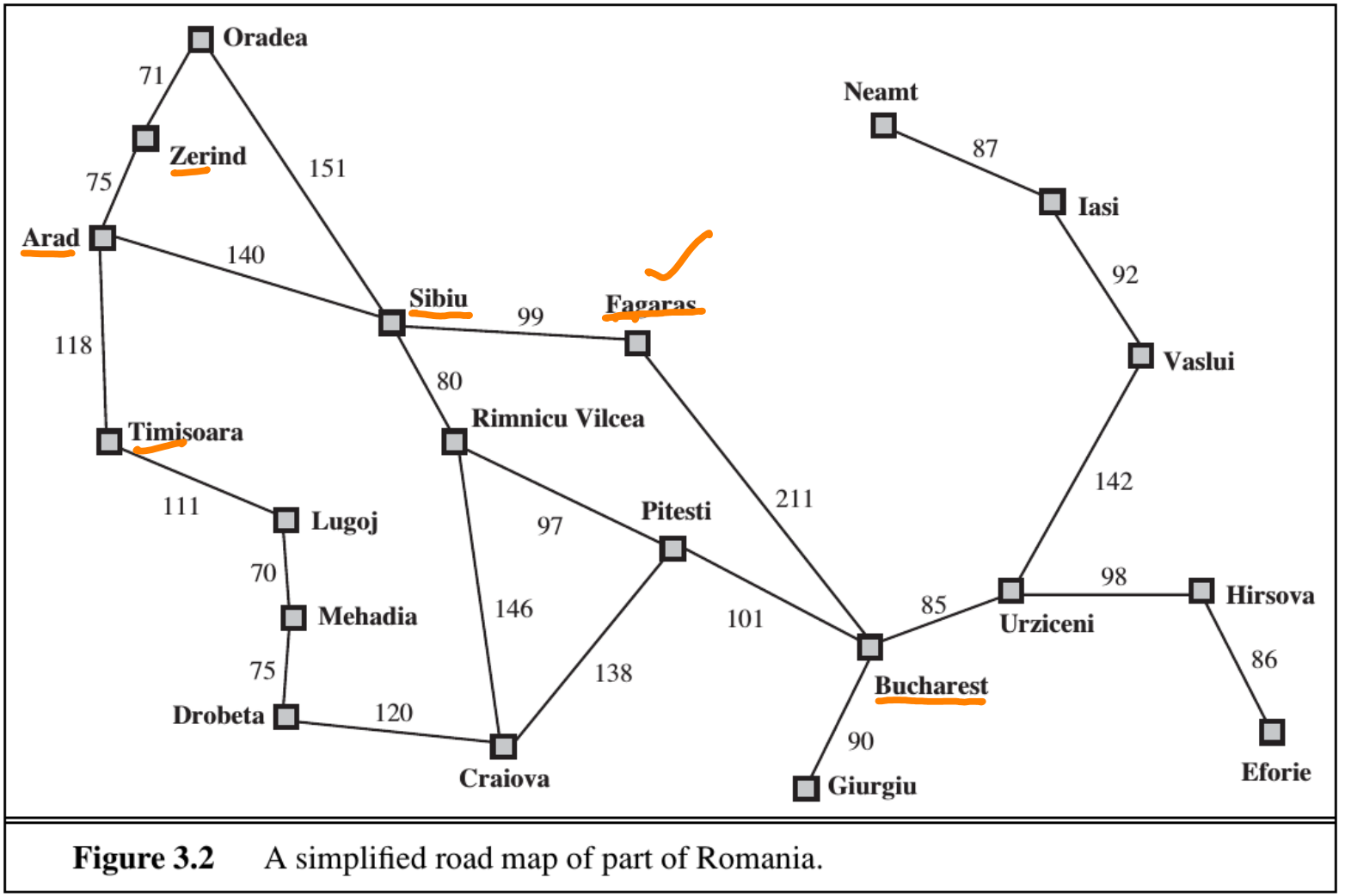



**Figure 3.2** A simplified road map of part of Romania.

# Greedy Best-first Search

- Tree search
- Graph search

# Greedy Best-first *Tree* Search

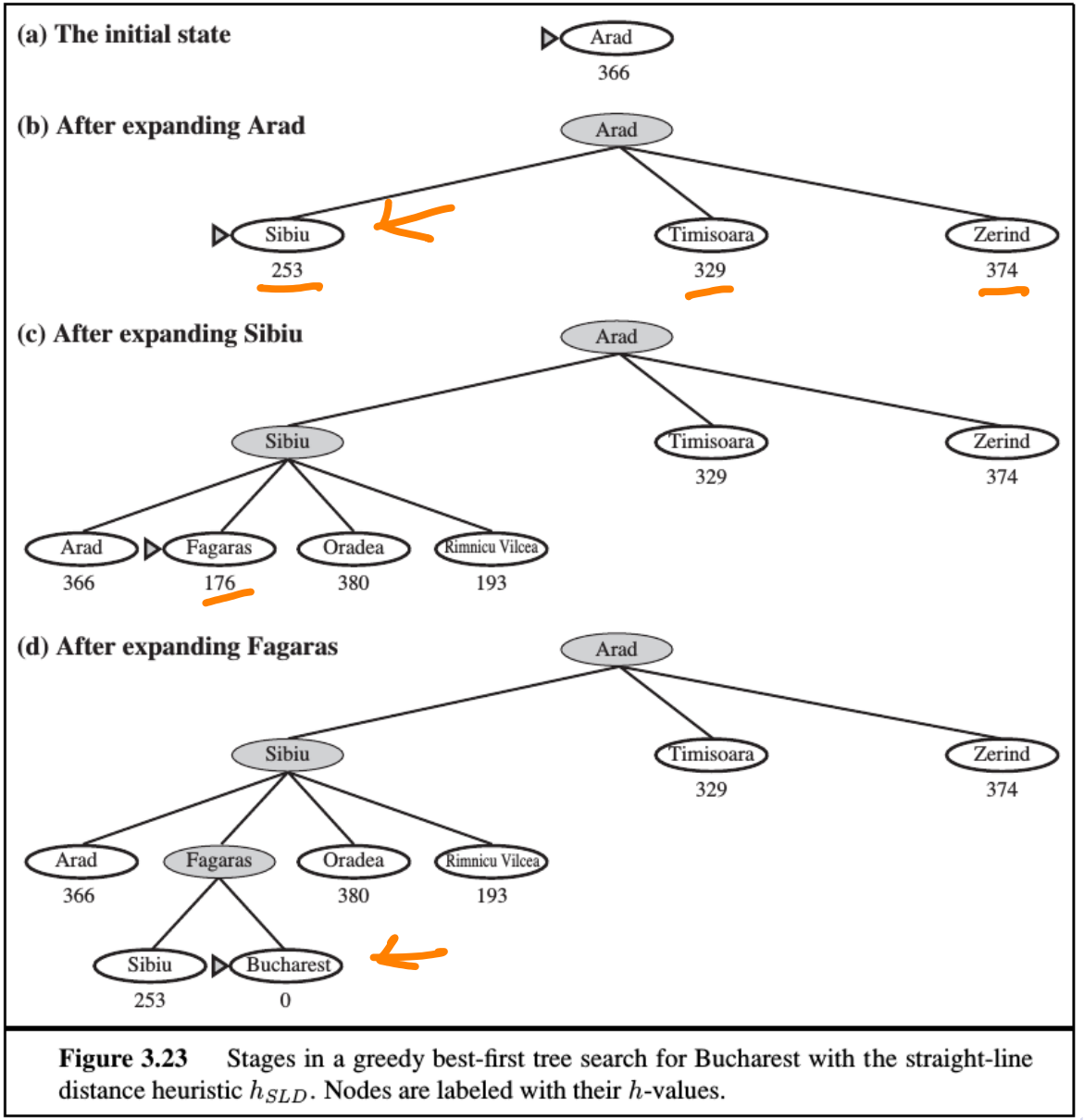



**Figure 3.23** Stages in a greedy best-first tree search for Bucharest with the straight-line distance heuristic $h_{SLD}$. Nodes are labeled with their $h$-values.

# Greedy Best-first Tree Search

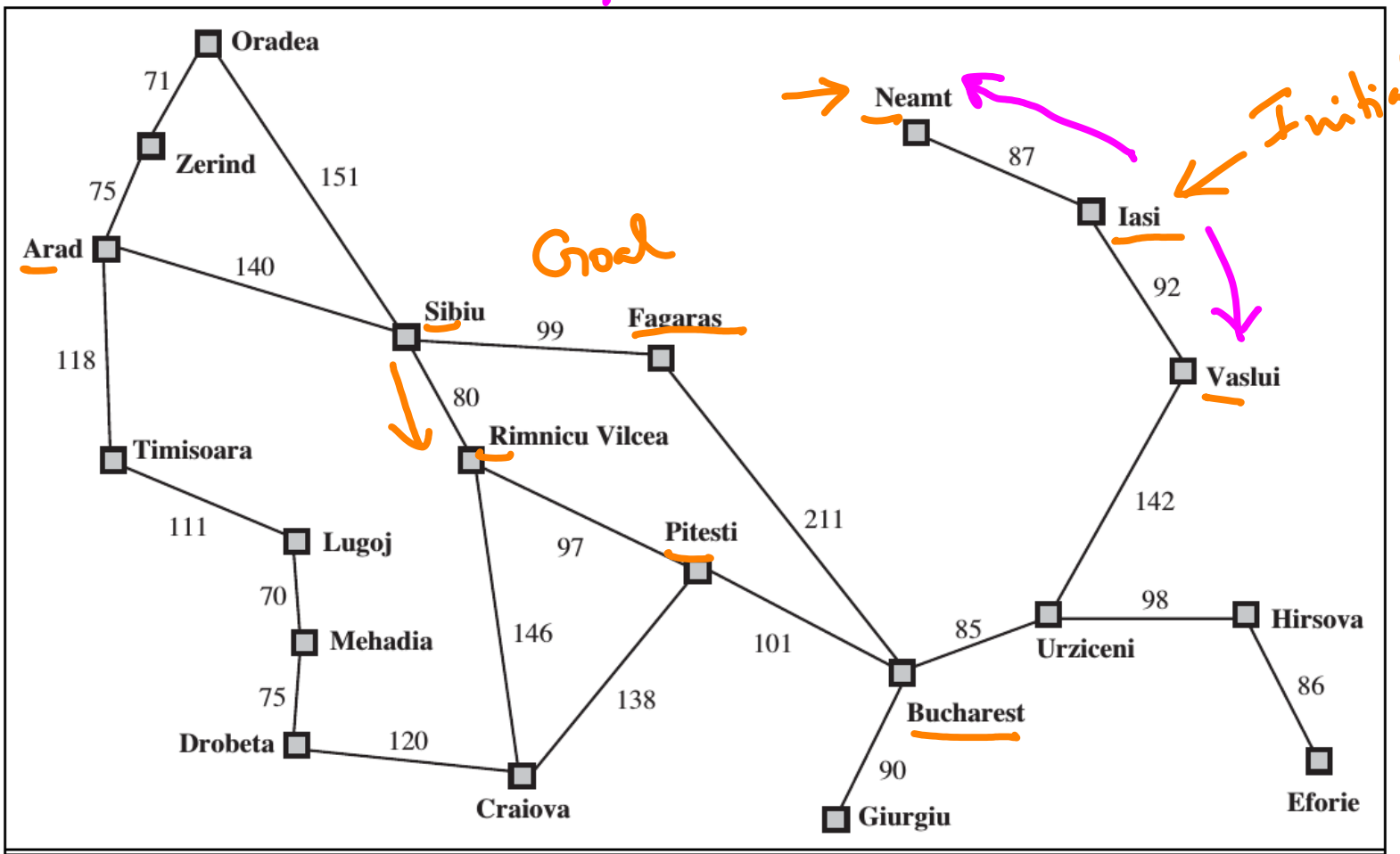



**Figure 3.2** A simplified road map of part of Romania.

# Performance evaluation

- Greedy Best-first Tree search for finite state space
  - Completeness?
  - Optimality?

# Performance evaluation

- Greedy Best-first Tree search for finite state space
  - Completeness?
  - Optimality?
  - Time complexity?

# Performance evaluation

- Greedy Best-first Tree search for finite state space
  - Completeness?
  - Optimality?
  - Time complexity? May not terminate.

# Performance evaluation

- Greedy Best-first Tree search for finite state space
  - Completeness?
  - Optimality?
  - Time complexity? May not terminate.
  - Space complexity?

# Performance evaluation

- Greedy Best-first Tree search for finite state space
  - Completeness?
  - Optimality?
  - Time complexity? May not terminate.
  - Space complexity? Linear in depth of the search tree.

# Performance evaluation

- Greedy Best-first Tree search for finite state space
  - Completeness?
  - Optimality?
  - Time complexity? May not terminate.
  - Space complexity? Linear in depth of the search tree.
- Greedy Best-first Graph search for finite state space
  - Completeness?
  - Optimality?

# Performance evaluation

- Greedy Best-first Tree search for finite state space
  - Completeness?
  - Optimality?
  - Time complexity? May not terminate.
  - Space complexity? Linear in depth of the search tree.
- Greedy Best-first Graph search for finite state space
  - Completeness?
  - Optimality?
  - Time complexity?

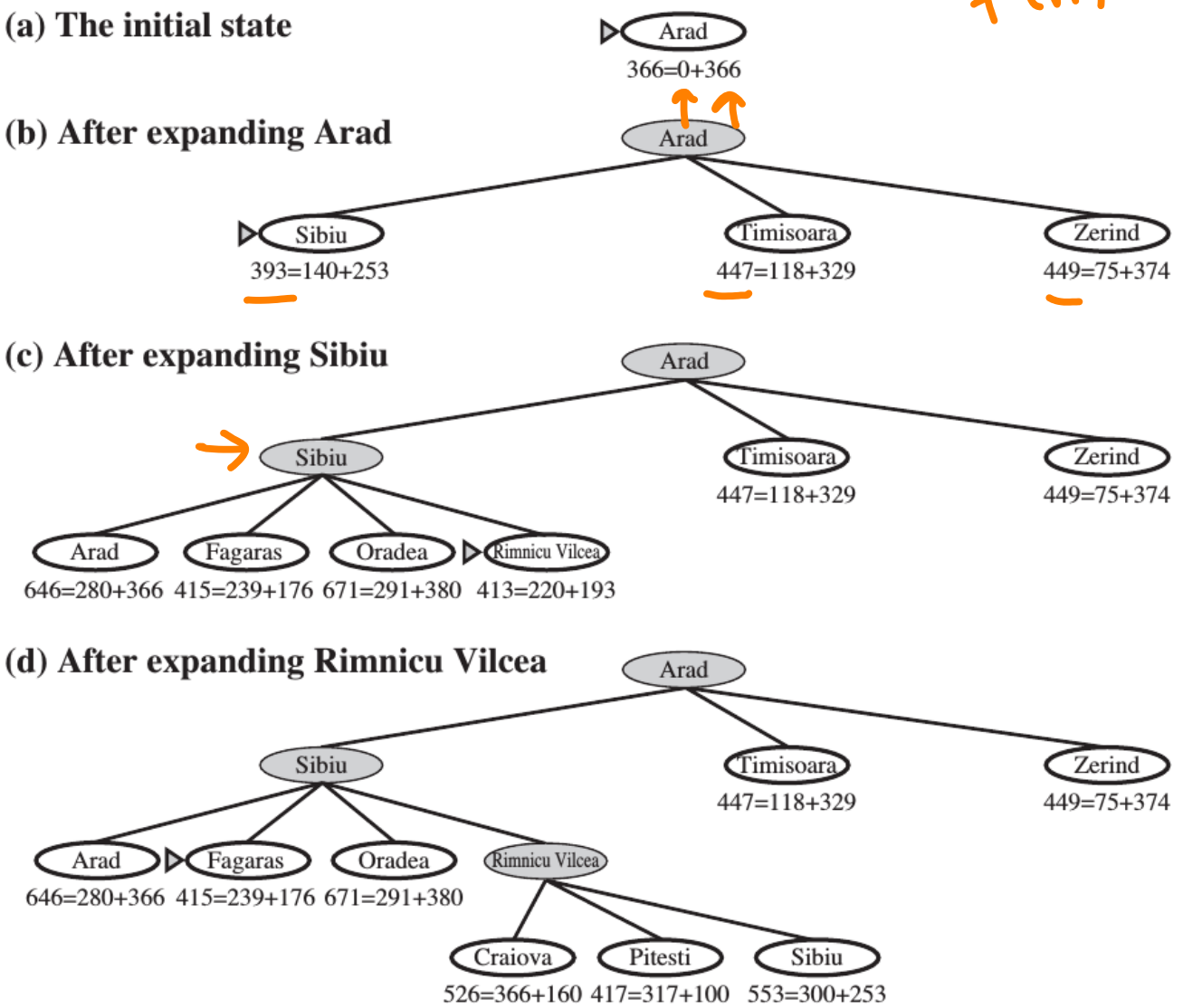
# Performance evaluation

- Greedy Best-first Tree search for finite state space
  - Completeness?
  - Optimality?
  - Time complexity? May not terminate.
  - Space complexity? Linear in depth of the search tree.
- Greedy Best-first Graph search for finite state space
  - Completeness?
  - Optimality?
  - Time complexity? $= O(b^m)$
  - Space complexity? $= O(b^m)$

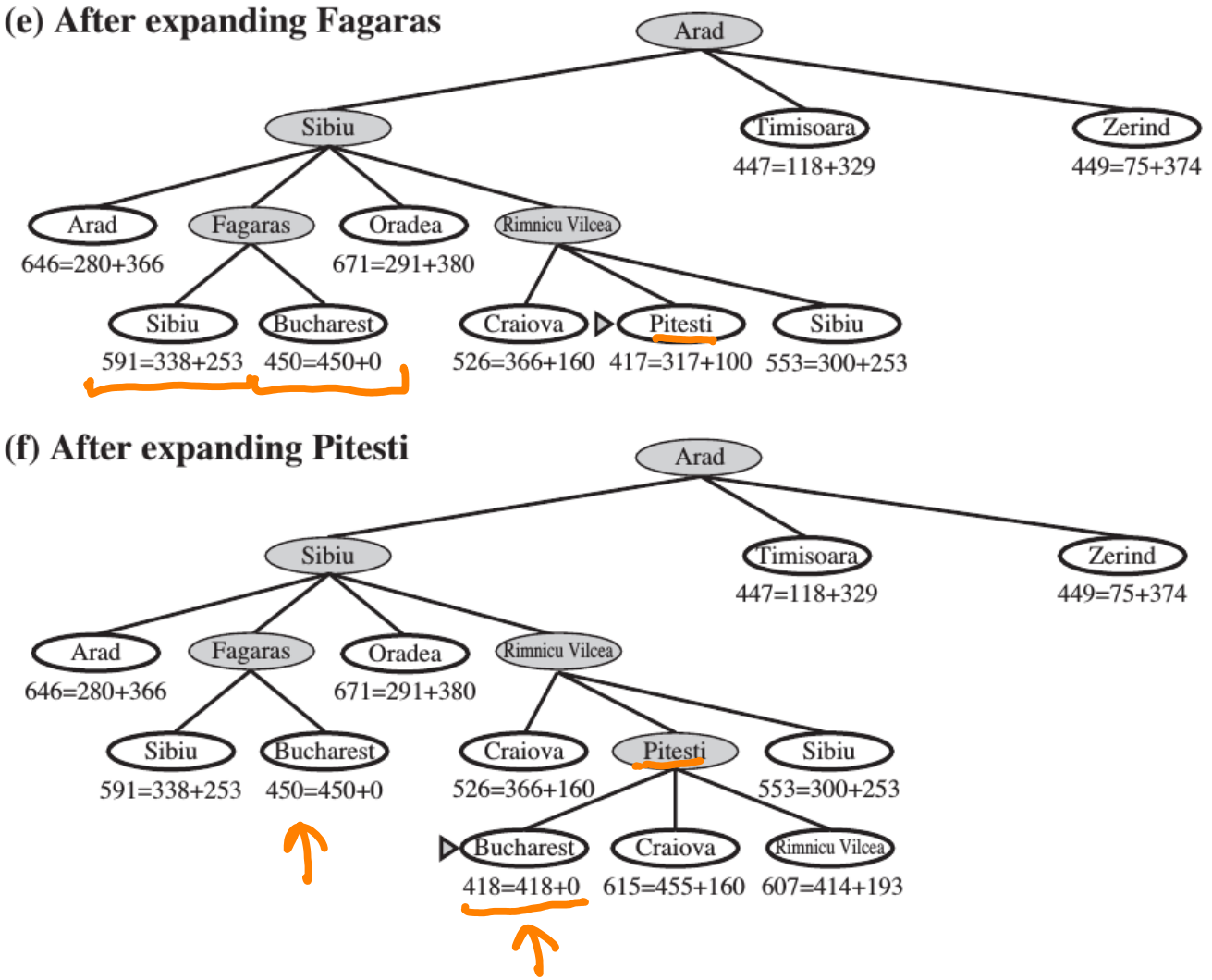
# A* search

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

# A* tree search



# A* tree search



# Uniform cost search vs. A* graph search

$g(n)$

$g(n)+h(n)$

$g(G)$

- Essentially, we have replaced the path function $g(n)$ in Uniform cost search with the evaluation function $f(n)$.

# Uniform cost search vs. A* graph search

$g(Goal)$

$f(Goal)$

$g(n)$

- Essentially, we have replaced the path function $g(n)$ in Uniform cost search with the evaluation function $f(n)$.
- Can we be sure that when the goal state is popped out of the priority queue in A* graph search the function $f(n)$ will be minimized?

$f(n)$

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

$= g(Goal)$

# Conditions for non-decreasing $f(n)$ in A$^*$ search

Consistent heuristic:

$$h(n) \leq c(n, a, n') + h(n')$$