Overview

- Heuristic Search Strategies
- The A* Algorithm (for Tree Search)
- Properties of A*
- Graph Search with A*
- (Good) Heuristics

Heuristic Functions

- heuristic function h: state space $\rightarrow \mathbb{R}$
- $h(n)$ = estimated cost of the cheapest path from node n to a goal node
- if n is a goal node then $h(n)$ must be 0
- heuristic function encodes problem-specific knowledge in a problem-independent way

Best-First Search

- an instance of the general tree search (or graph) search algorithm
  - strategy: select next node based on an evaluation function *f*: state space $\rightarrow \mathbb{R}$
  - select node with lowest value $f(n)$
- implementation:
  selectFrom(*fringe*, *strategy*)
  - priority queue: maintains fringe in ascending order of *f*-values

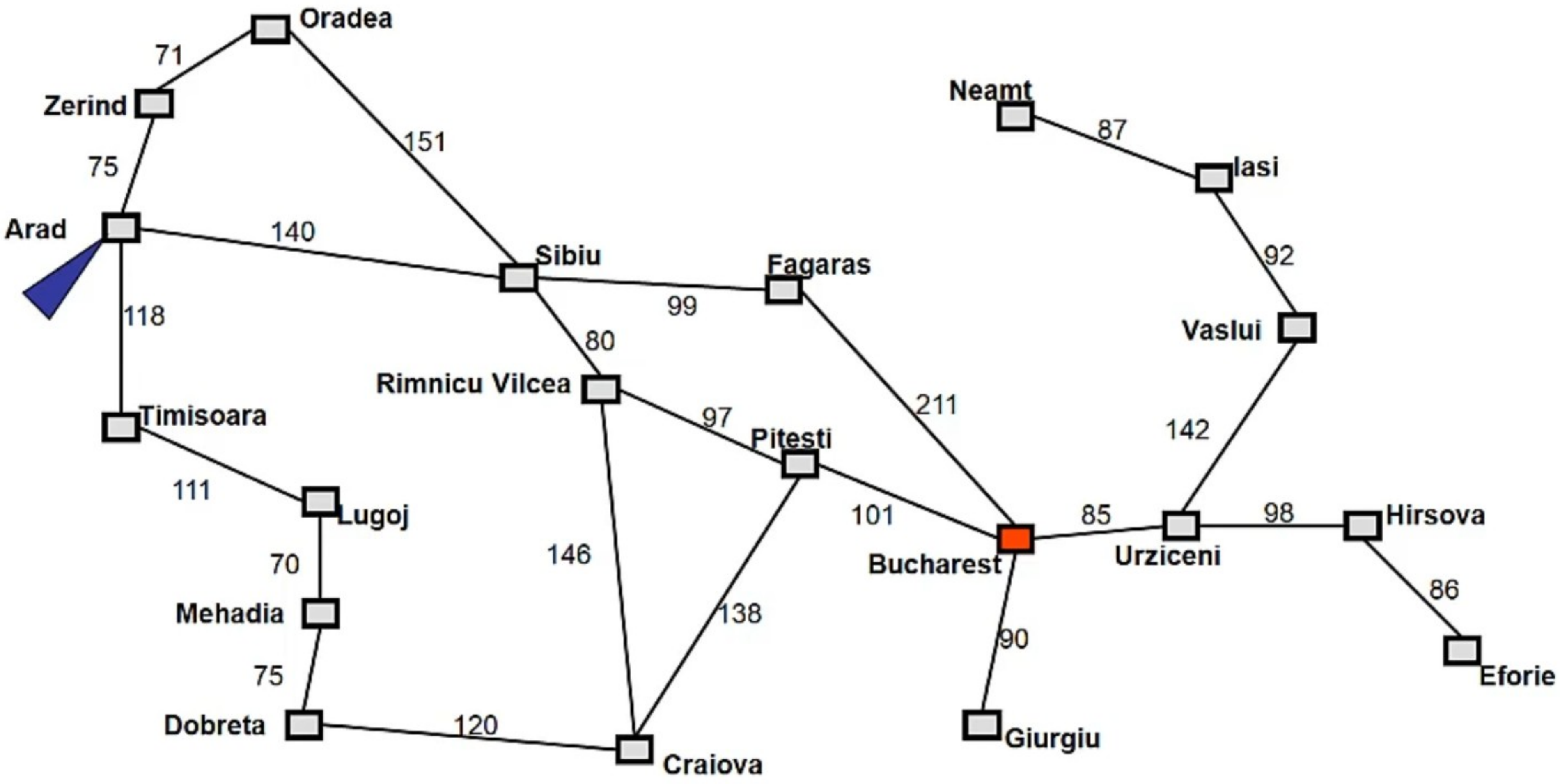
Greedy Best-First Search

- use heuristic function as evaluation function:

$$f(n) = h(n)$$

  - always expands the node that is closest to the goal node
  - eats the largest chunk out of the remaining distance, hence, "greedy"

Real-World Problem: Touring in Romania

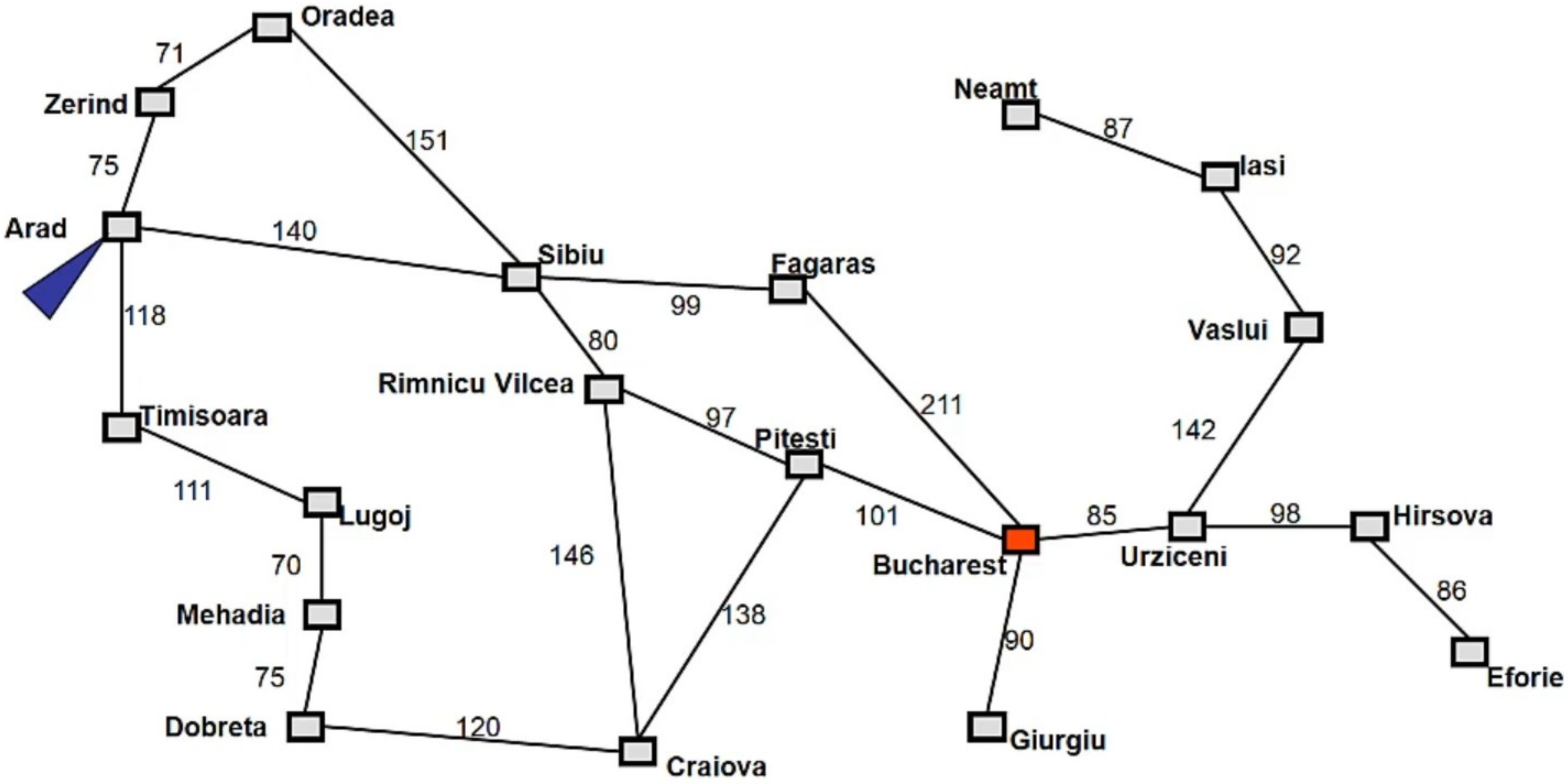


Touring Rumania : Heuristics

$h_{SLD}(n)$ = Straight line distance to Bucharest

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

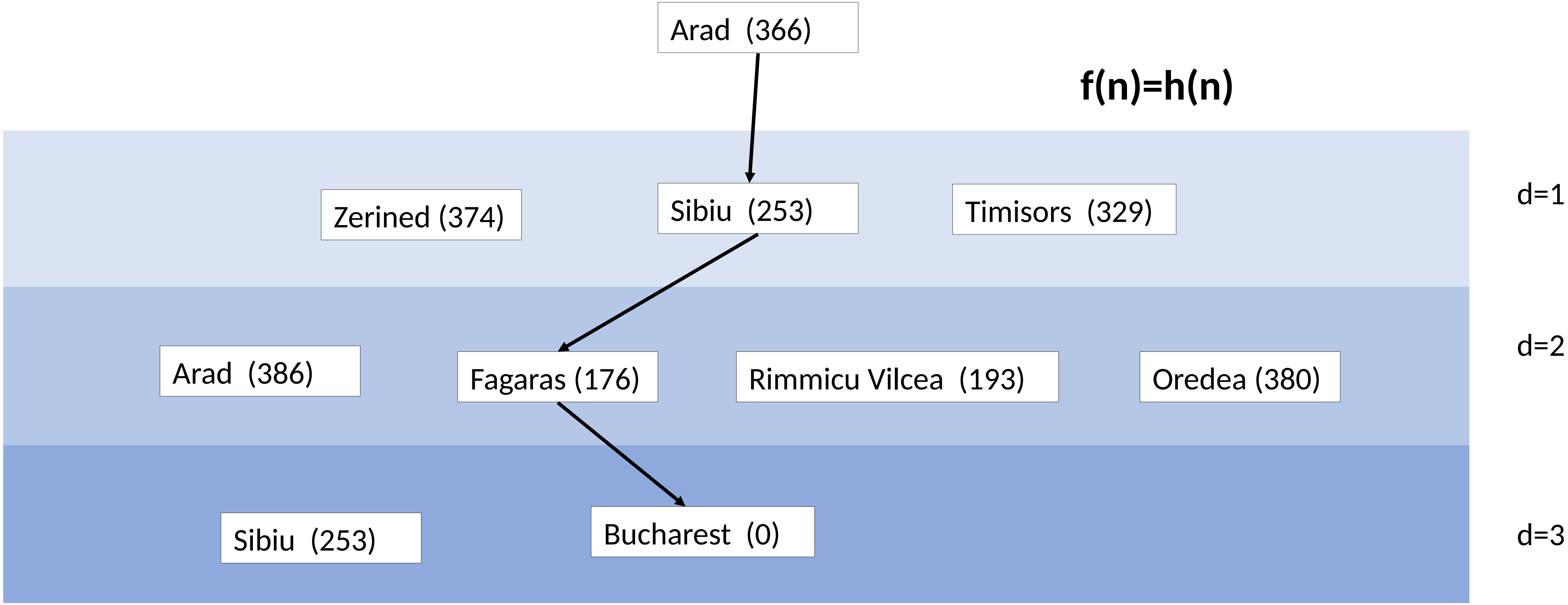
Real-World Problem: Touring in Romania



Greedy Best First Search

selected

$f(n)=h(n)$



Total Distance = 450

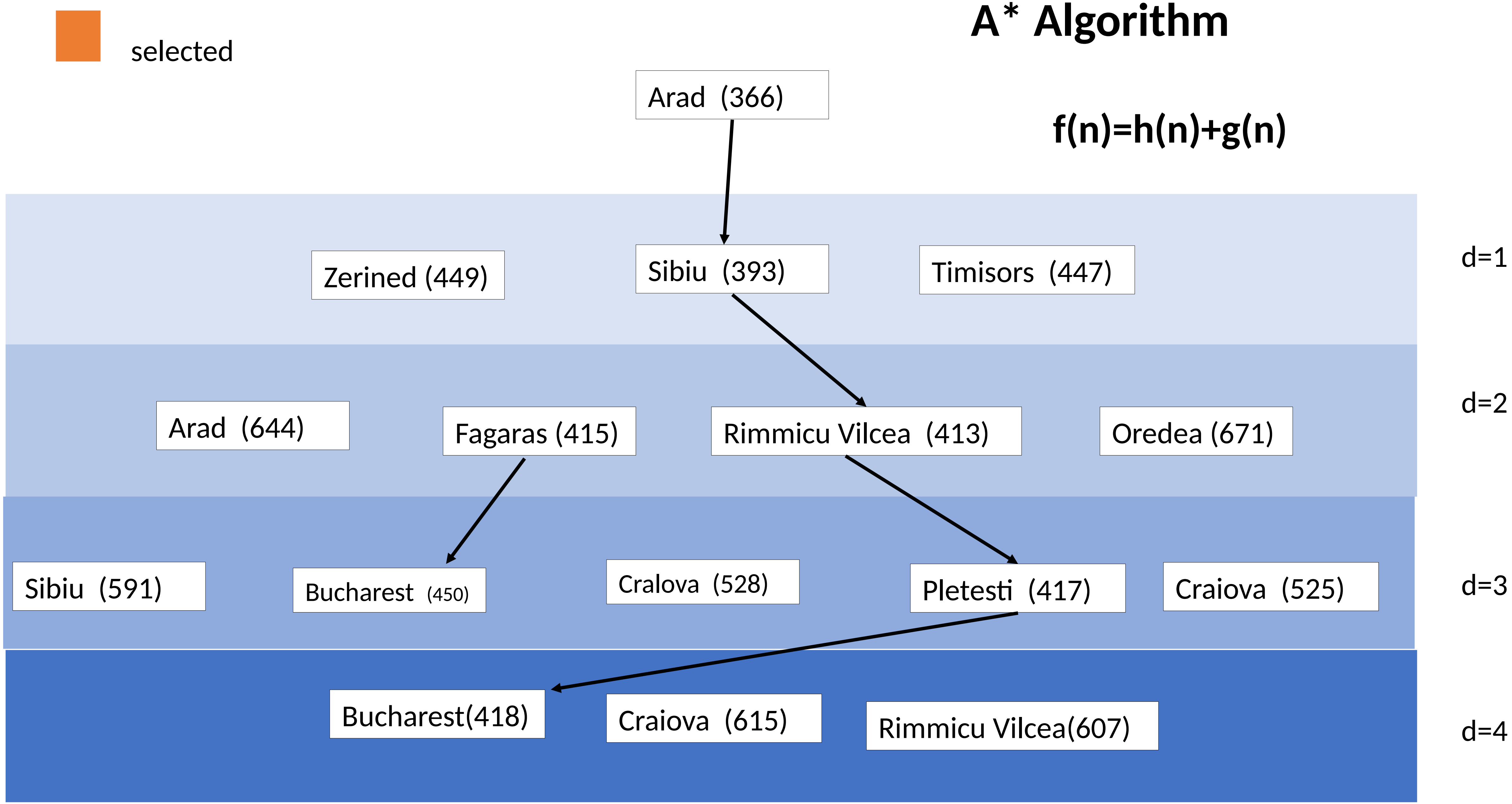
A* Search

- best-first search where
  $$f(n) = h(n) + g(n)$$
  – $h(n)$ the heuristic function (as before)
  – $g(n)$ the cost to reach the node n
- evaluation function:
  $f(n)$ = estimated cost of the cheapest solution through n

A* is optimal if h(n) is admissible

A* Algorithm

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



Total Distance = 418

8-Tile Puzzle

The tiles can be moved to the empty space horizontal and vertical directions only.

- Exercise : Find the plan to solve this problem using A* algorithm.

- g(n) =Depth of the search tree
- (i)   h(n) = Number of misplaced tiles
- (ii) h(n) = Manhattan distance to the goal.

Initial State

1	**2**	**3**
	4	6
7	5	8

Goal State

1	**2**	**3**
4	5	6
7	8	

Admissible Heuristics

A heuristic $h(n)$ is admissible if it *never overestimates* the distance from n to the nearest goal node.

- example: h_{SLD}
- A* search: If $h(n)$ is admissible then $f(n)$ never overestimates the true cost of a solution through n.

$h(n) \leq$ actual distance to the goal from n

Completeness of A*: Contours

- <u>contours</u>: sets of states that can be reached within a certain cost
  - prerequisite for drawing contours: *f*-values along a path are non-decreasing
- A* fans out from the start node, adding nodes in concentric bands (contours) of increasing *f*-values
- A* is complete: it must reach a contour that includes a goal node