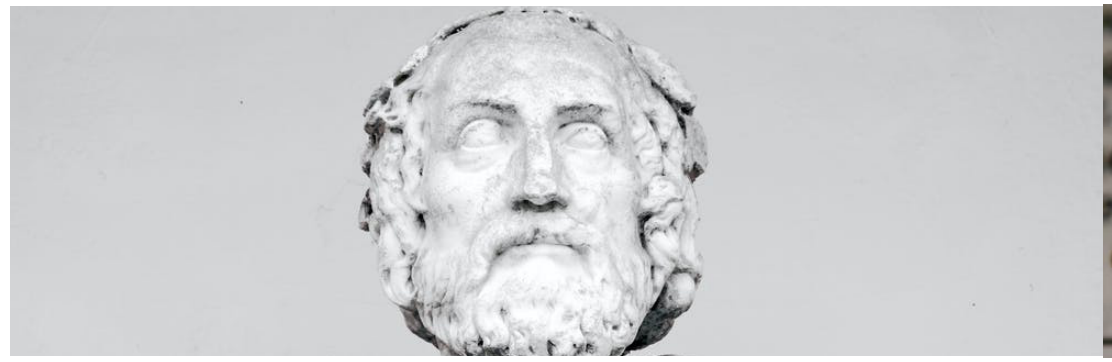
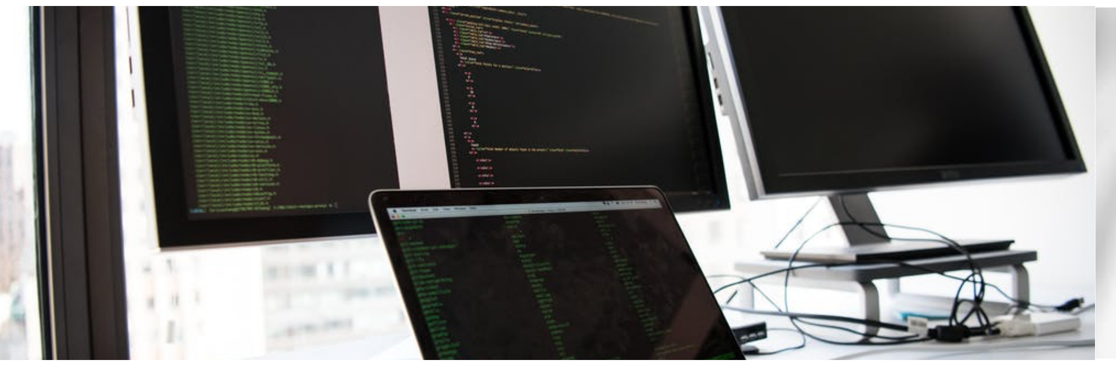
COMPSCI 761: ADVANCED TOPICS IN ARTIFICIAL INTELLIGENCE

SUMMARY I

Anna Trofimova, July 2022

FOUNDATIONS OF AI




- Philosophy (428 B.C — present) - What is knowledge? What is mind?
- Mathematics (c. 800 — present) - Formal rules, Probabilities
- Psychology (1879 — present) - How do humans think?
- Linguistics (1957 — present) - Ho does language relate to thoughts?
- Computer engineering (1940 — present) - Efficient computers
- Biocybernetics (1940’s — present) - How brains process information?

PERSPECTIVES ON AI DEFINITION

Thinking humanly	Thinking Rationally
"The exciting new effort to make computers think . . . *machines with minds*, in the full and literal sense." (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning . . ." (Bellman, 1978)	"The study of mental faculties through the use of computational models." (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)
Acting Humanly	**Acting Rationally**
"The art of creating machines that perform functions that require intelligence when performed by people." (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)	"Computational Intelligence is the study of the design of intelligent agents." (Poole et al., 1998) "AI . . . is concerned with intelligent behavior in artefacts." (Nilsson, 1998)

PERSPECTIVES ON AI DEFINITION

> *The term AI contains an explicit reference to the notion of intelligence. However, since intelligence (both in machines and in humans) is a* ***vague*** *concept, although it has been studied at length by psychologists, biologists, and neuroscientists, AI researchers use mostly the notion of* ***rationality****. This refers to the ability to choose* ***the best action*** *to take in order to achieve a certain* ***goal****, given certain* ***criteria*** *to be optimized and the available resources. Of course, rationality is not the only ingredient in the concept of intelligence, but it is a significant part of it.*

High-Level Expert Group on Artificial Intelligence, "A definition of AI: Main capabilities and scientific disciplines.", https://digital-strategy.ec.europa.eu/en/library/definition-artificial-intelligence-main-capabilities-and-scientific-disciplines (2019).

EXAMPLES OF RATIONAL AGENTS

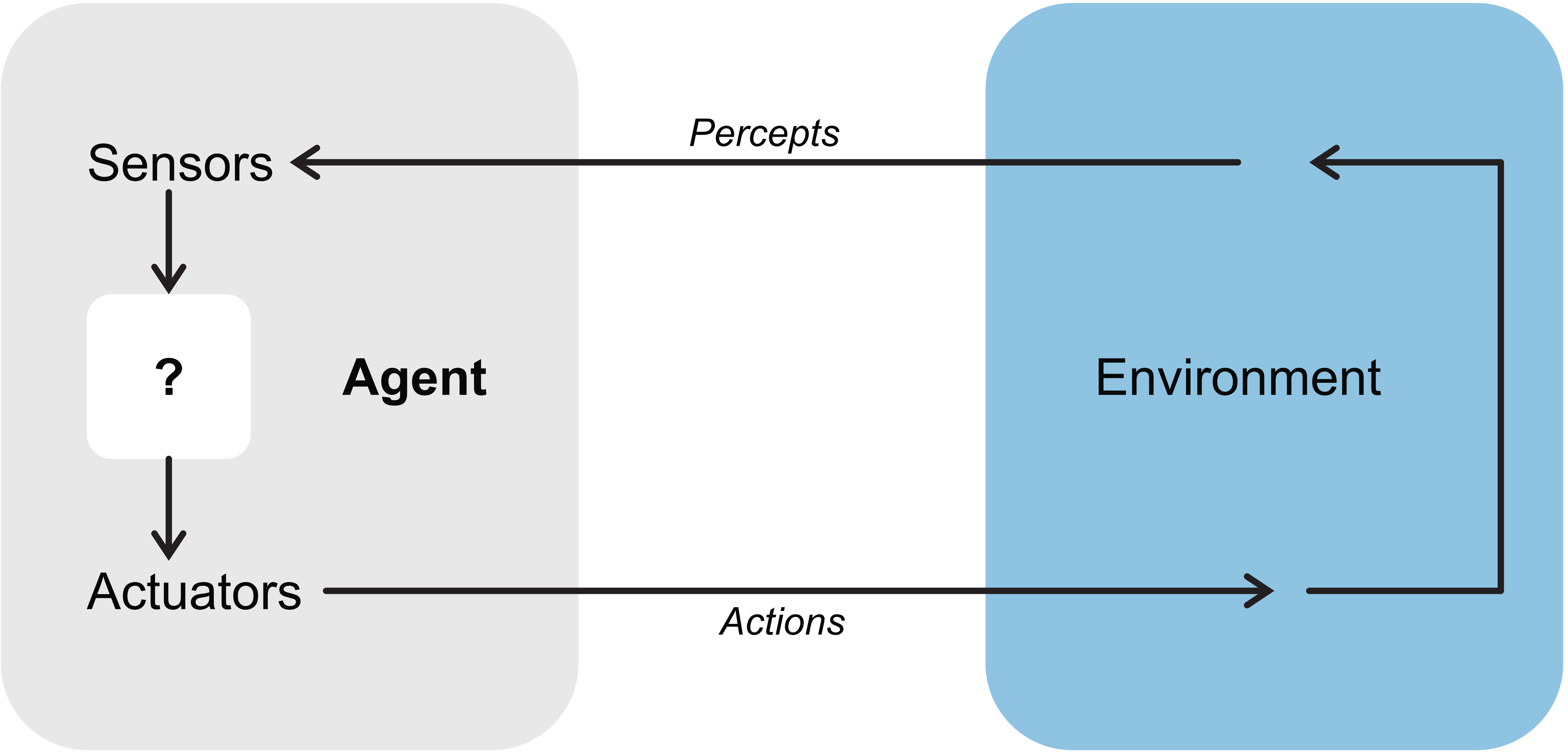
People
Examples: teacher, stock trader, doctor

Animals & Bacterias
Examples: birds, worms, wolves, bacillus

Computers/Devices
Examples: smart cars, games, advising systems

Organisations
Examples: Government, UoA, ant colonies

AGENT MODEL



EXAMPLE OF PAGE DESCRIPTION ★

Agent	Percepts	Actions	Goal	Environment
Medical diagnosis system	Symptoms, findings, patient's answers	Ask questions, run tests, prescribe treatments	Healthy patients, minimise cost	Patient, hospital
Chess playing system	Position of the chess pieces	Move a chess pieces	Score	Chess board, chess pieces
Movie recommendation system	Preferences, watched movies, feedback	Suggest/rank movies, gather feedback	Client's happiness, watched movies	Movie streaming service
Refinery controller	Temperature, pressure readings	Open and close valves, adjust temperatures	Maximise purity, safety	Refinery

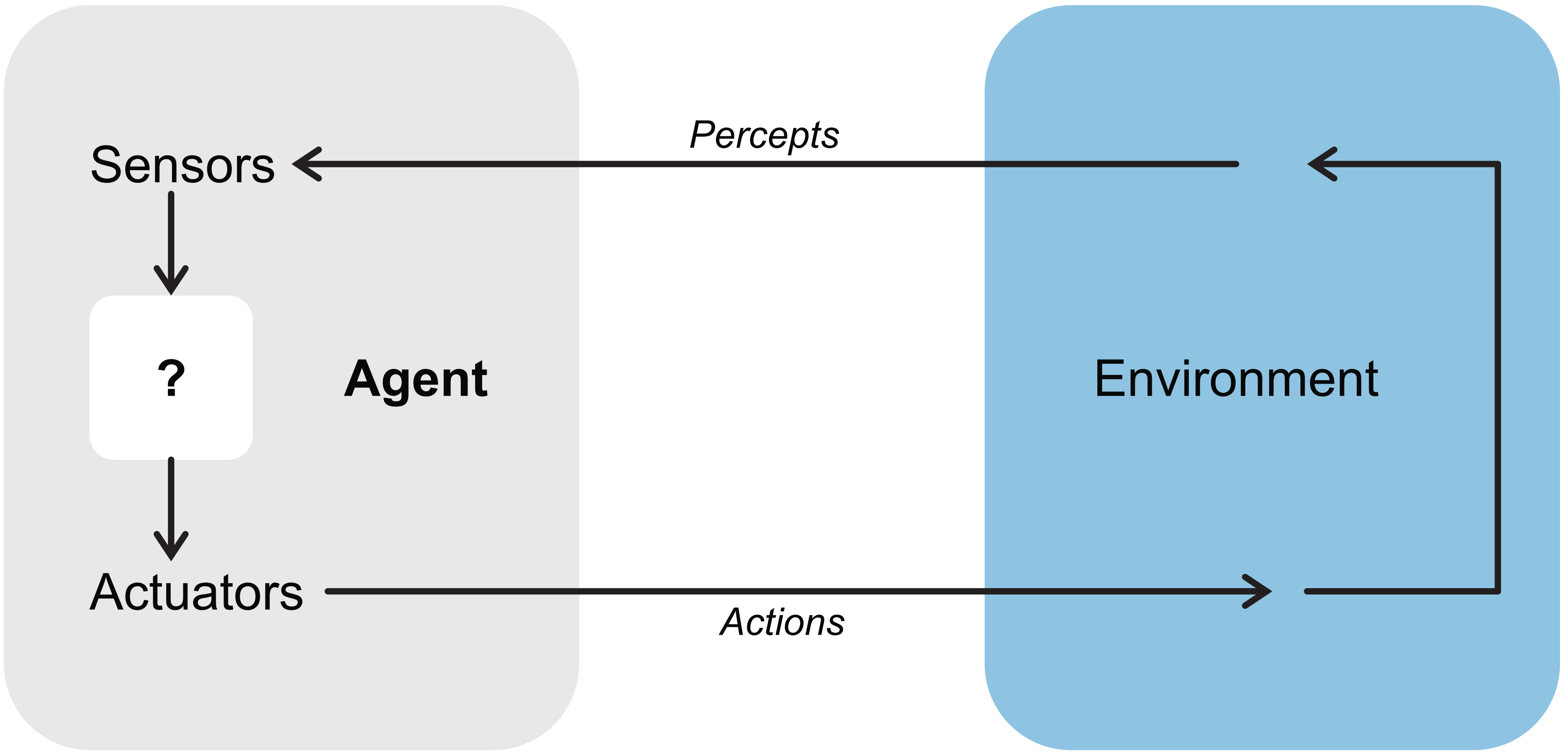
SPECIFYING AND CLASSIFYING TASK ENVIRONMENT

We want a unified framework that can be used to specify, characterise, compare, and contrast different AI tasks.

Tasks are the problems to which rational agents are the solutions.

To design a rational agent, we must specify the task environment:

- **P**erformance measure
- **E**nvironment
- **A**ctuators
- **S**ensors



PEAS MODEL

- Performance measure
  How can we tell if an agent is doing a good or bad job?
  What are the qualities of performance that we would like to measure?
- Environment
  What are the components/attributes of the environment (which are relevant to the agent)?
- Actuators
  What are the outputs that enable action upon an environment?
- Sensors
  What are the inputs that sense and provide data for the agent?

EXAMPLE: PEAS DESCRIPTION ★

- Use PEAS description for a automatic taxi driver.
    - Performance measure?
    - Environment?
    - Actuators?
    - Sensors?

CLASSIFYING ENVIRONMENT TYPE

Simulated vs Situated/Embodied

Static vs Dynamic

Discrete vs Continuous

Fully Observable vs Partially Observable

Deterministic vs Stochastic

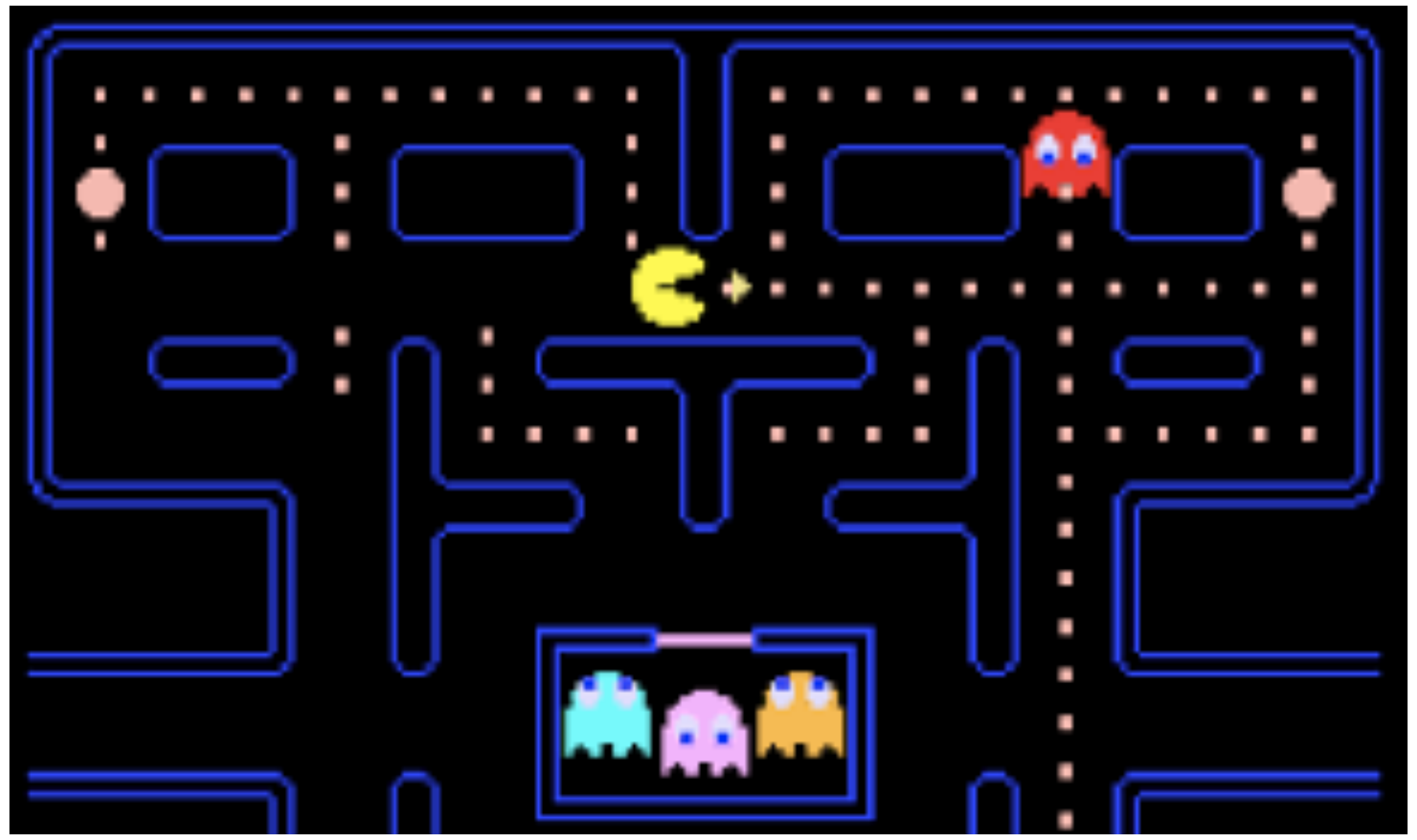
Episodic vs Sequential

Known vs Unknown

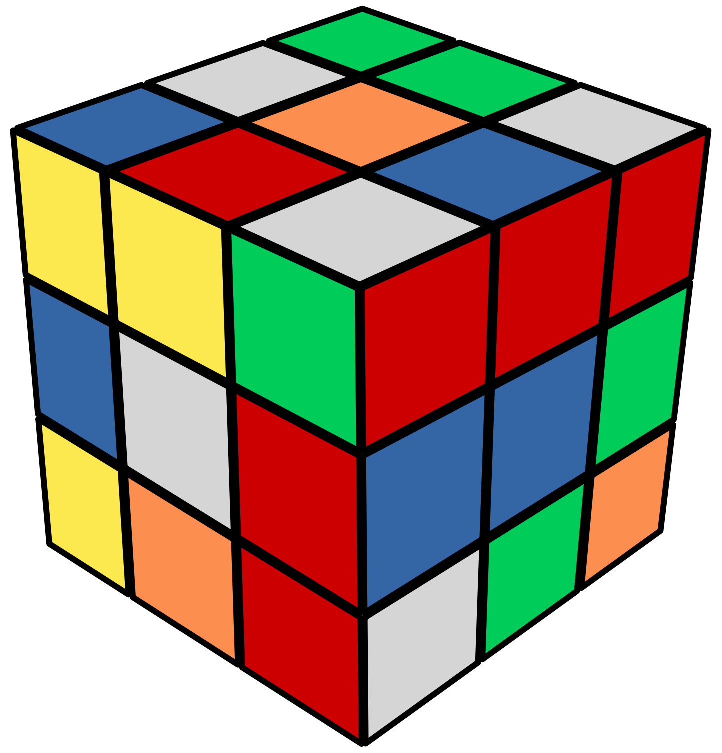
Single-Agent vs Multi-Agent

EXAMPLE: CLASSIFYING ENVIRONMENT TYPE ★

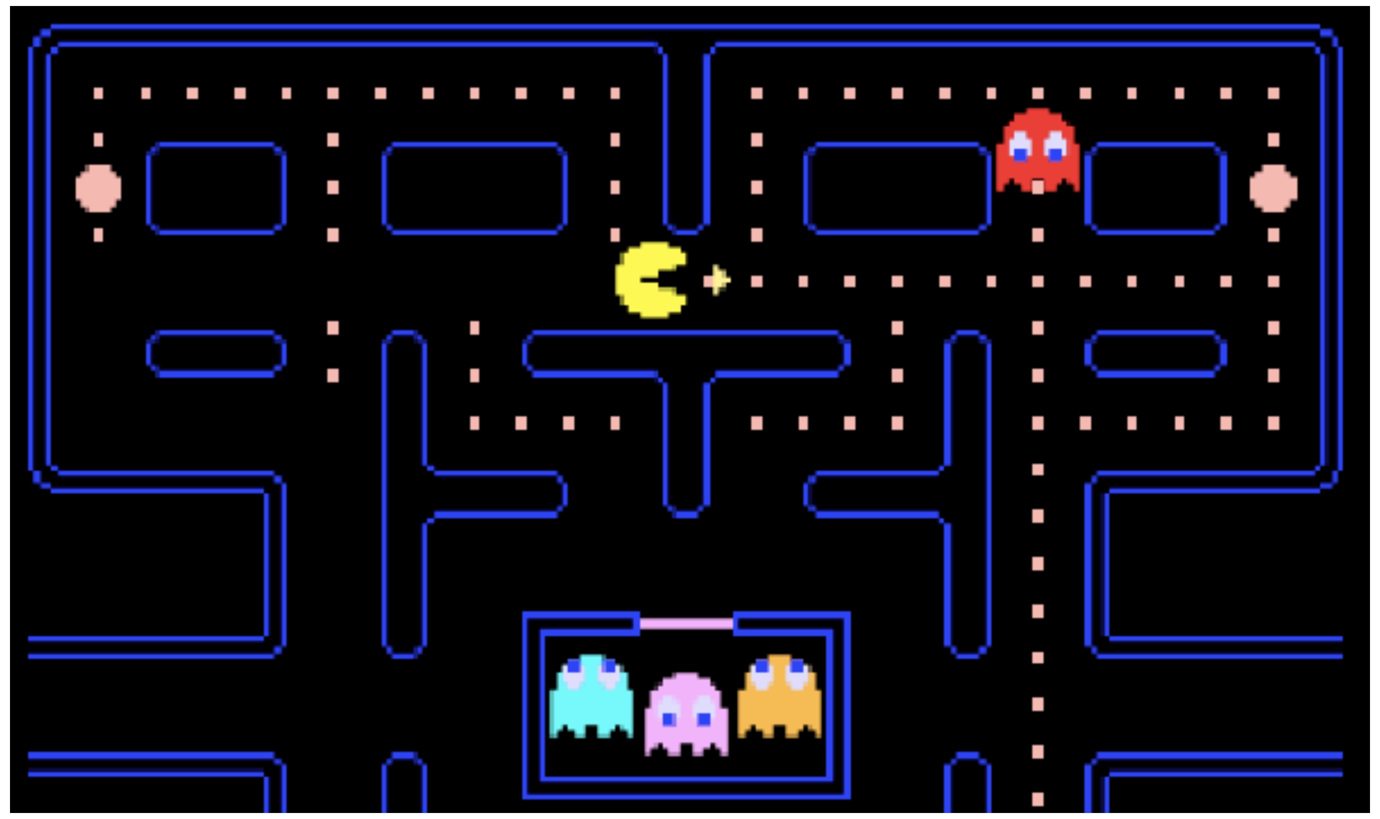
- Pick the environment description that can be applied to a game of pacman:
  - a. Simulated, fully observable, discrete,
  - b. Situated, fully observable, discrete
  - c. Simulated, partially observable, discrete
  - d. Simulated, fully observable, continuous



SEARCH PROBLEM VS CSP VS GAME



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



STATE SPACE PROBLEM

A state-space problem consists of:

- a set of states
- a subset of states called the **start states**
- a set of actions
- an **action function**: given a state and an action, returns a new state
- a (set of) goal state, specified as Boolean function, goal(s)
- a criterion that specifies the quality of an acceptable solution.

EXAMPLE: SPECIFYING A STATE-SPACE PROBLEM ★

- Specify a state-space problem for a Rubik's cube:
  - A set of states: all possible positions of the 26 coloured blocks
  - The start state: the initial position of the coloured blocks
    - [b1, …, b26 = (x26, y26, r26)], where each block b_i = (x, y, r)
  - Set of actions: rotation of the faces
    - $f(x_i, y_i, r_i) = (x_{i\_new}, y_{i\_new}, r_{i\_new})$
  - The goal state: each side of the cube being a single colour
    - [b1 = (1, 1, 1), b2 = (1, 2, 1), …)
- Which algorithm would you use to search for a solution? What is path cost then?

SOLVING PROBLEMS BY SEARCHING

- Search as a “weak method” of problem solving with wide applicability

- Uninformed search methods (use no problem-specific information)
  - Uninformed (or “blind”) search strategies use only the information available in the problem definition (can only distinguish a goal from a non-goal state)

- Informed search methods (use heuristics to improve efficiency)
  - Informed (or “heuristic”) search strategies use task-specific knowledge.

SEARCH TREE

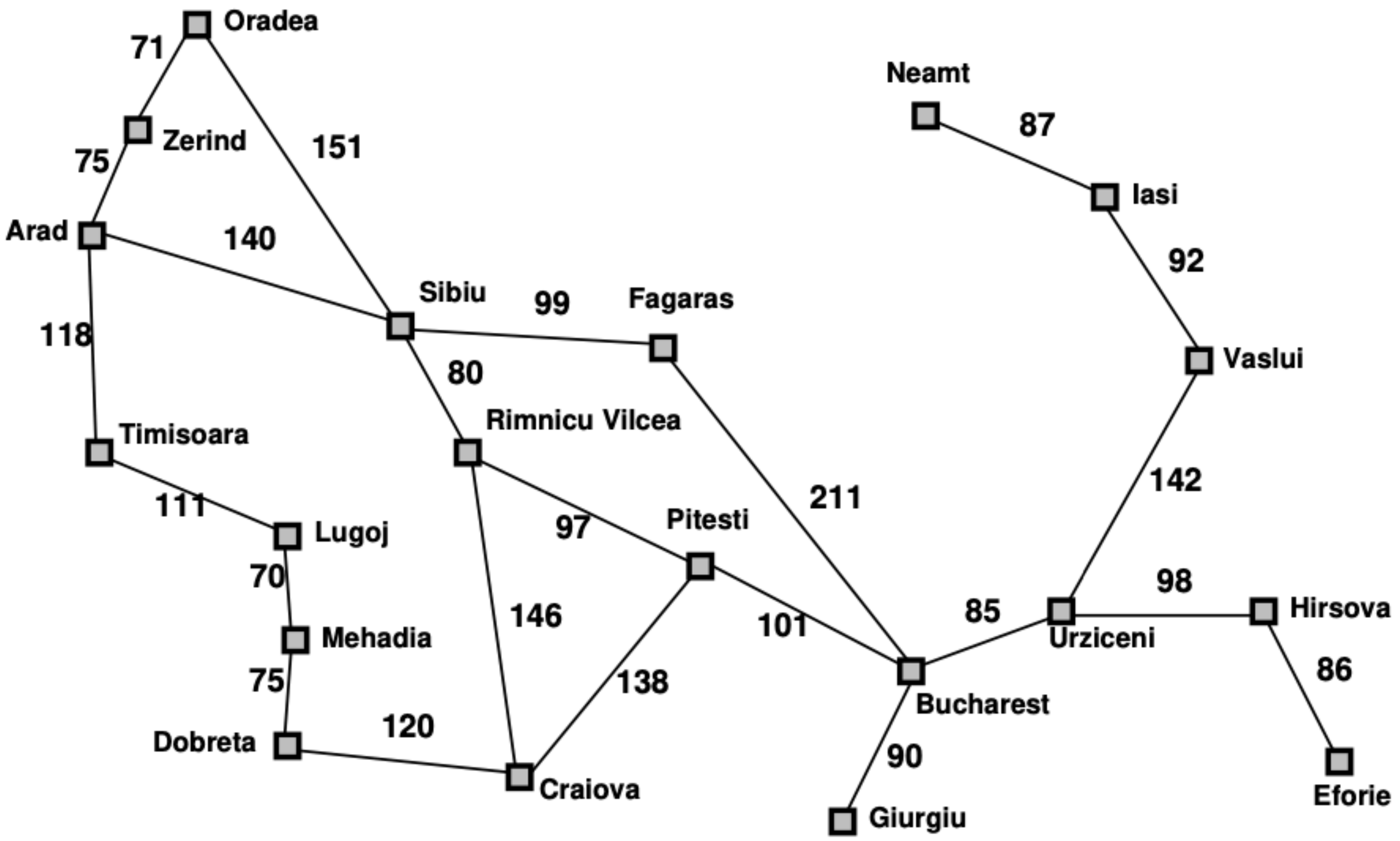
- Search tree: superimposed over the state space (which can be presented as a state-space graph).
- Root: search node corresponding to the initial state.
- Leaf nodes: correspond to states that have no successors in the tree because they were not expanded or generated no new nodes.

COMPLEXITY RESULTS FOR AN UNINFORMED SEARCH

Algorithm	Completeness	Admissibility	Space	Time
Breadth-First Search	guaranteed, if b is finite	guaranteed, if arcs have the same cost and >=0	$O(b^d)$	$O(b^d)$
Depth-First Search	not guaranteed	not guaranteed	O(bm)	$O(b^m)$
Uniform Cost Search	guaranteed, if b is finite and cost ≥ ε	guaranteed	$O(b^{[1 + C / \varepsilon]})$	$O(b^{[1 + C / \varepsilon]})$
Depth Limited Search	guaranteed, if b is finite	not guaranteed	O(bk)	$O(b^k)$
Iterative Deepening Search	guaranteed, if b is finite	guaranteed, if arcs have the same cost and >=0	O(bd)	$O(b^d)$
Bidirectional Search	depends on search algorithms used	depends on search algorithms used	$O(b^{d/2})$	$O(b^{d/2})$

EXAMPLE OF UNIFORMED SEARCH QUESTION★

- If you use Uniform Cost Search algorithm to find a path from Arad to Bucharest, what would be the 4th expanded node considering that the generated nodes are added to the frontier in the alphabetical order?



F = [*Arad(0)]
1st: F.pop() = Arad
F = [*Zerind(75), *Timisoara(118), *Sibiu(140)]
2nd: F.pop() = Zerind
F = [Timisoara(118), Sibiu(140), *Oradea(146), *Arad(150)]
3rd: F.pop() = Timisoara
F = [Sibiu(140), Oradea(146), Arad(150), *Arad(236), *Lugoj(229)]
4th: F.pop() = Sibiu

COMPLEXITY RESULTS FOR INFORMED SEARCH

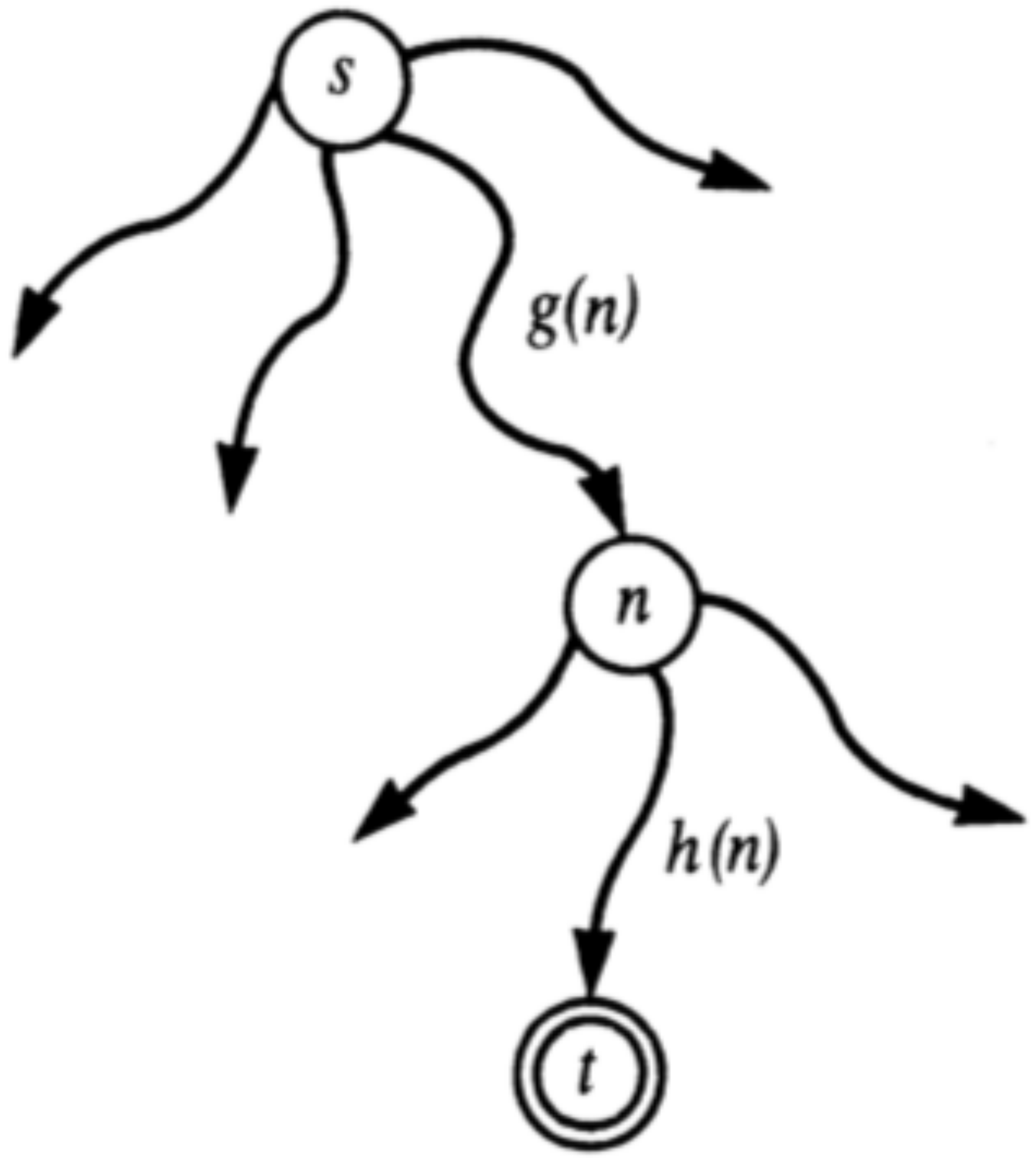
Algorithm	Completeness	Admissibility	Space	Time
Greedy Best-First Search	guaranteed, if b is finite and with repeated-state checking	not guaranteed	$O(b^m)$	$O(b^m)$
A* Search	guaranteed	guaranteed if heuristic is admissible/consistent	$O(b^d)$	$O(b^d)$
Iterative Deepening A* Search	guaranteed	guaranteed if heuristic is admissible/consistent	$O(bd)$	$O(b^d)$

HEURISTIC FUNCTION

Heuristic estimate *f(n) of* the cost of the cheapest paths from **s** to **t** via **n**: f(n) = g(n)+ h(n)

g(n) is an estimate of the cost of an optimal path from s to n

h(n) is an estimate of the cost of an optimal path from n to t.



A* SEARCH – CONDITIONS FOR OPTIMALITY

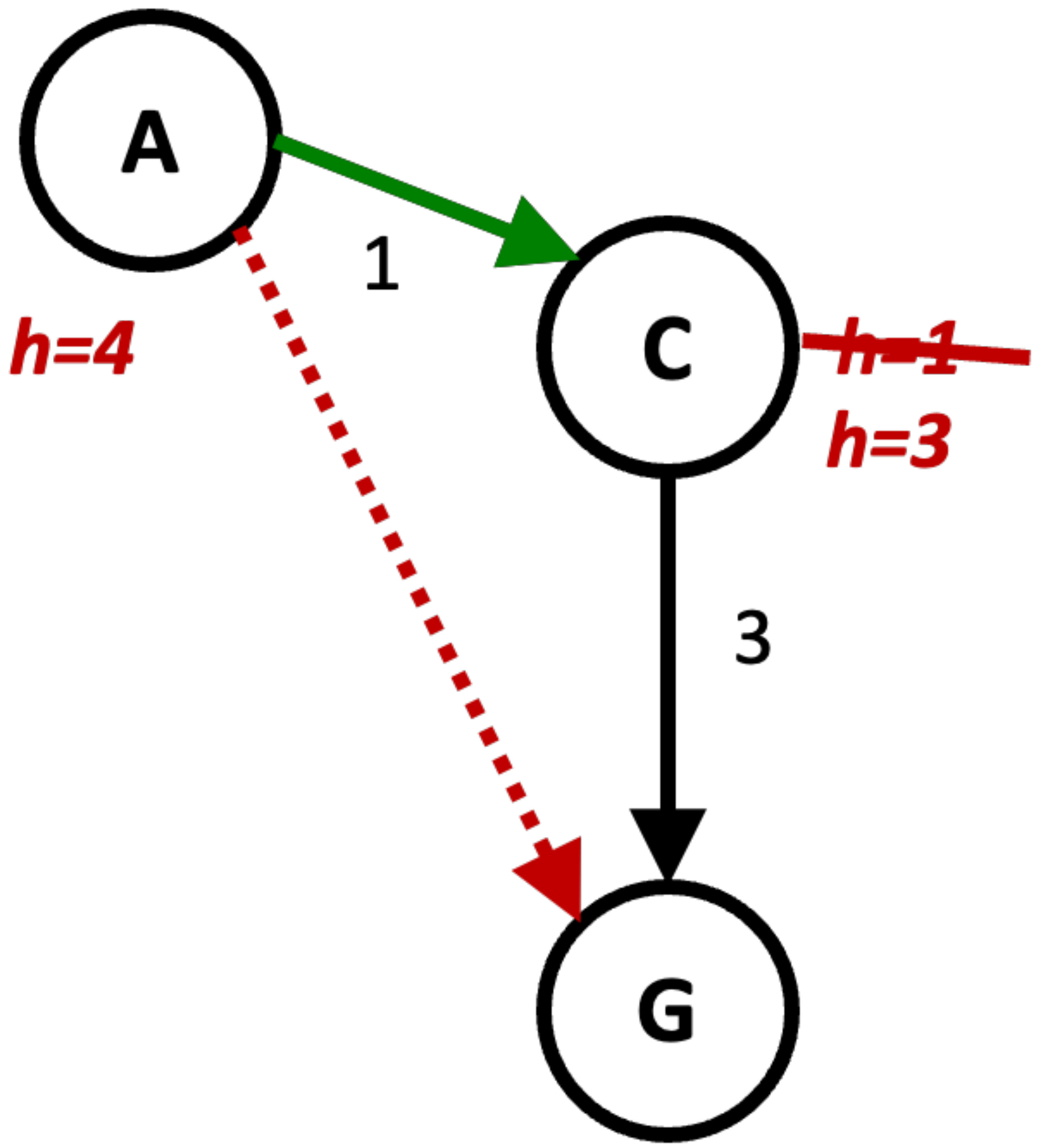
- Heuristic ***h*** is called **admissible** if
- $\forall n\ h(n) \leq h^*(n)$ where $h^*(n)$ is true cost from n to the goal

- If ***h*** is **admissible** then ***f (n)*** never overestimates the actual cost of the best solution through *n.*

- Example: $\boldsymbol{h_{SLD}}$ is admissible for delivery robot problem because the shortest path between any two points is a line.

- Theorem: A* Search finds optimal solution if ***h*(n)** is admissible.

CONSISTENCY OF HEURISTIC



Admissibility: heuristic cost ≤ actual cost to goal

Consistency: heuristic "edge" cost ≤ actual cost for each edge

$|h(A) - h(C)| \leq c(A, C)$

Or $h(C) - c(A, C) \leq h(A) \leq c(A, C) + h(C)$ (triangle inequality)

Consequences of consistency:

- The f value along a path never decreases:

$h(A) \leq c(A, C) + h(C) \Rightarrow g(A) + h(A) \leq g(A) + c(A, C) + h(C)$

- A* graph search is optimal (no-reopenings)

WHAT DOES h_1 BETTER THAN h_2 MEANS

- What we'd like h_1 better than h_2 for problem P to mean is that A* solving P using h_1 takes less time/memory than A* using h_2
- However, that is hard to predict in general.
- So, instead we look at node expansions as a proxy for time/memory
- If A* expands fewer nodes using h_1 than when using h_2 we say it is "better"
- Happily, we can characterise when this is likely to happen, namely when h_1's values are not lower than h_2's values.

TRIVIAL HEURISTIC, DOMINANCE

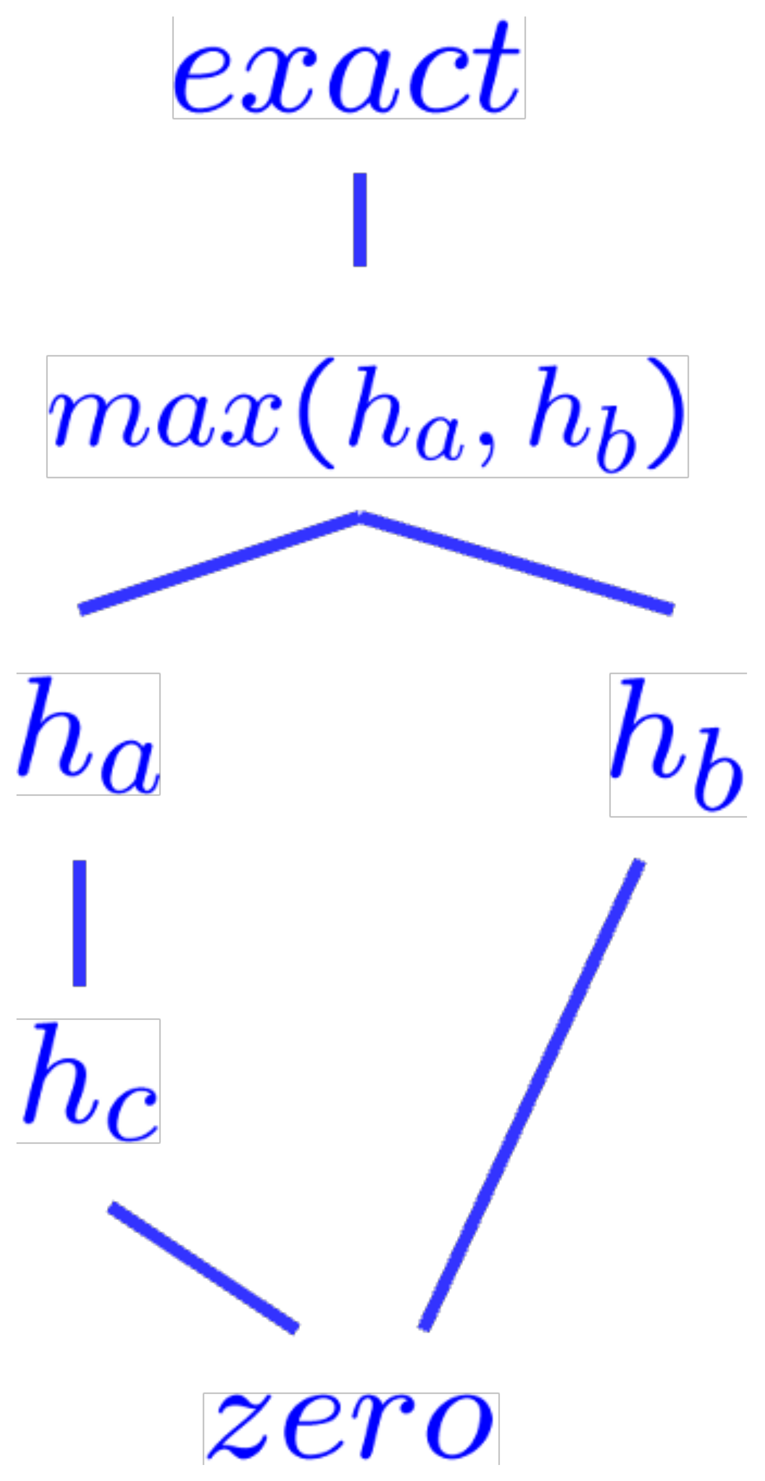
- Dominance: if $h_a(n) \geq h_c(n)$ for all *n* (both admissible) then *h*2 dominates *h*1 and is better for search. So the aim is to make the heuristic *h*() as large as possible, but without exceeding $h^*()$.

  $h_a \geq h_c$ if $\forall n$: $h_a(n) \geq h_c(n)$

- Heuristics form a semi-lattice:
  - Max of admissible heuristics is admissible

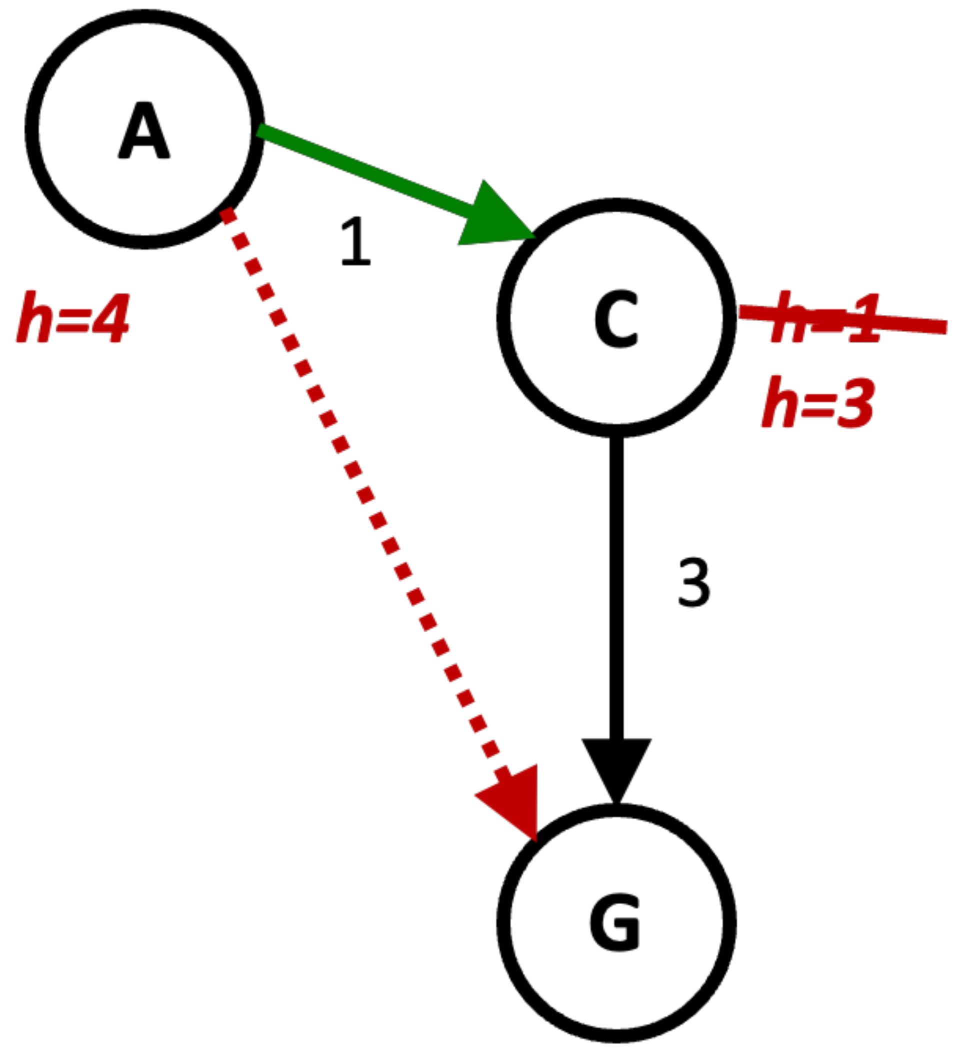
  $h(n) = max(\ h_a(n), h_b(n)\ )$

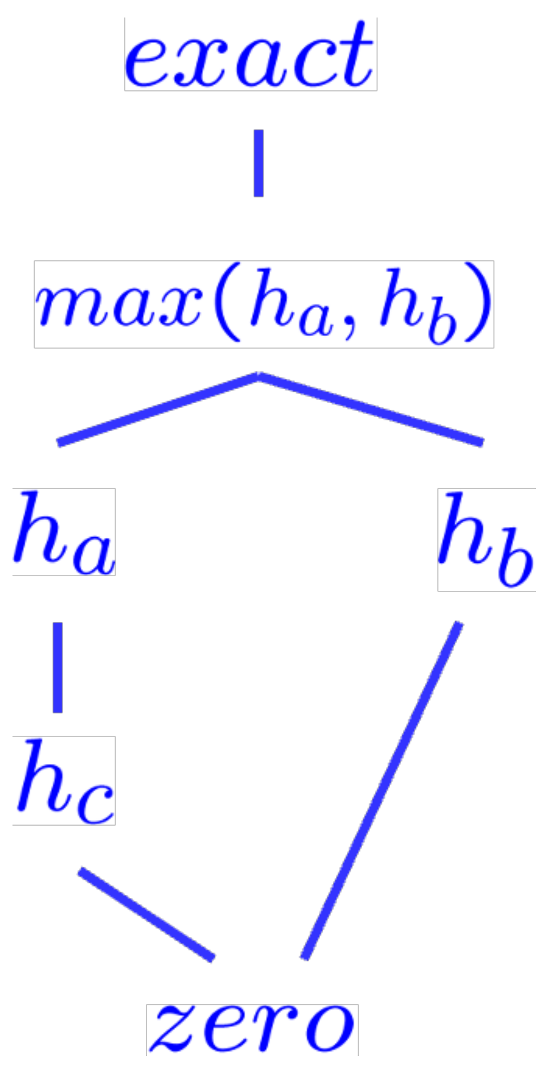
- Trivial heuristics
  - Bottom of lattice is the zero heuristic (what does this give us?)
  - Top of lattice is the exact heuristic



A QUESTION ABOUT HEURISTICS ★

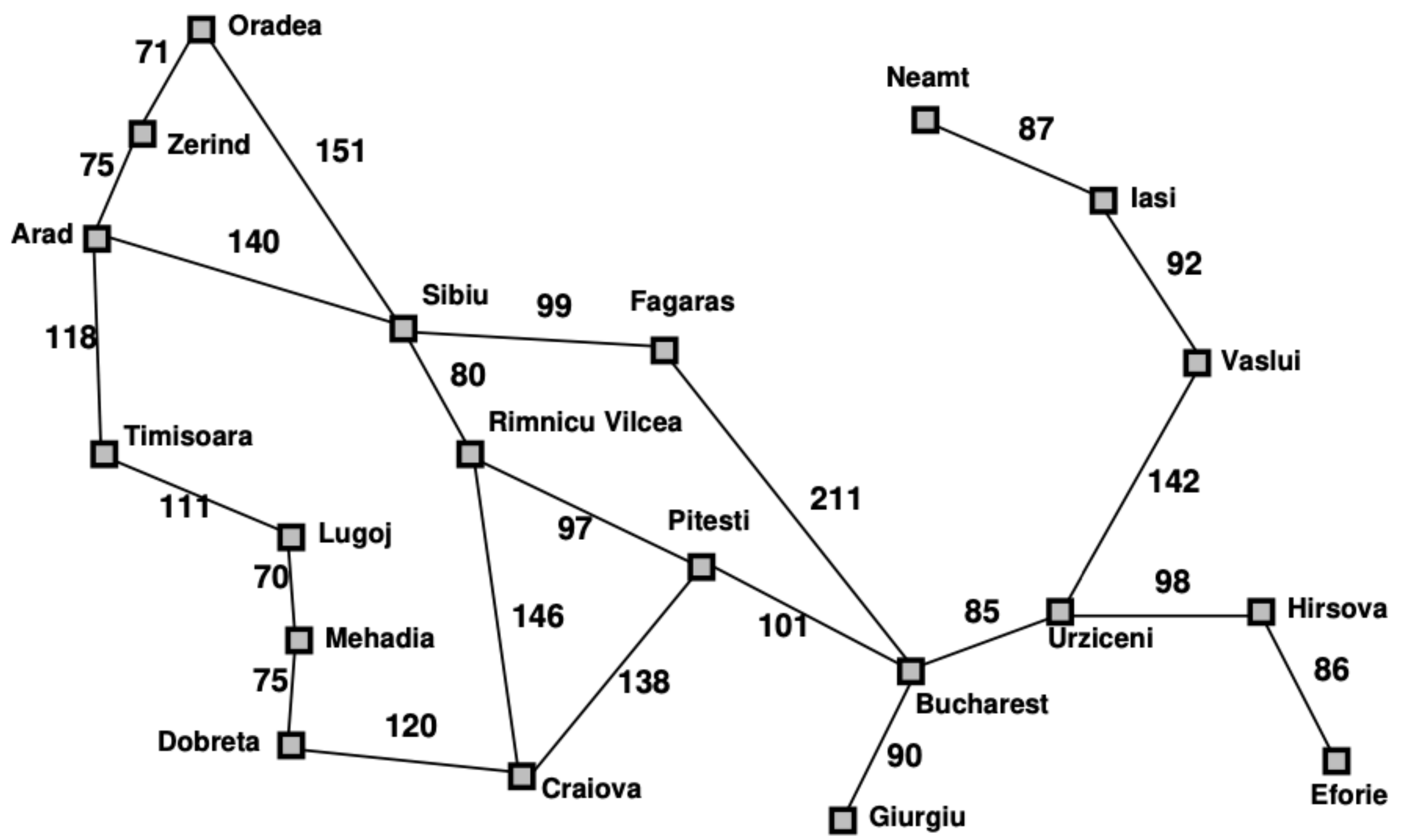
- No example :(
- Implicit or explicit question about heuristic properties.





EXAMPLE OF AN INFORMED SEARCH QUESTION★

- If you use Greedy Search algorithm to find a path from Arad to Bucharest, what would be the 4th expanded node considering that the generated nodes are added to the frontier in the alphabetical order?



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

F = [*Arad (336)]
1st: F.pop() = Arad
F = [*Sibiu(253), *Zerind(374), *Timisoara(329)]
2nd: F.pop() = Sibiu
F = [*Fagaras(178), *Rimnicu Vilcea (193), Zerind(374), Timisoara(329), *Arad(366)]
3rd: F.pop() = Fagaras
F = [*Bucharest(0), Rimnicu Vilcea (193), *Sibiu(253), Zerind(374), Timisoara(329), Arad(366)]
4th: F.pop() = Bucharest

* meaning added this iteration