AI 2002
Artificial Intelligence

Dr. Hashim Yasin

Informed (Heuristic) Search Strategies

Best-First Search

- A node is selected for expansion based on an **evaluation function**, $\boldsymbol{f(n)}$.
  - the *lowest* evaluation is expanded first
- Best-first algorithms include as a component of f a **heuristic function**, denoted $\boldsymbol{h(n)}$.
- $\boldsymbol{h(n)}$ = ***estimated*** cost of the cheapest path
  - $\boldsymbol{h(n)}$ takes a *node* as **input**, but, unlike $\boldsymbol{g(n)}$, it depends only on the *state* at that node.

$$\boldsymbol{f(n) = h(n)}$$

A* Search

- We **can bias Uniform-cost search** to find the shortest path to the goal.
- In fact, we are interested in by using a **heuristic function $h(n)$** which is an estimate of the distance from a state to the goal.
- It evaluates nodes by combining $g(n)$, the cost to reach the node, and $h(n)$, the cost to get from the node to the goal:

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

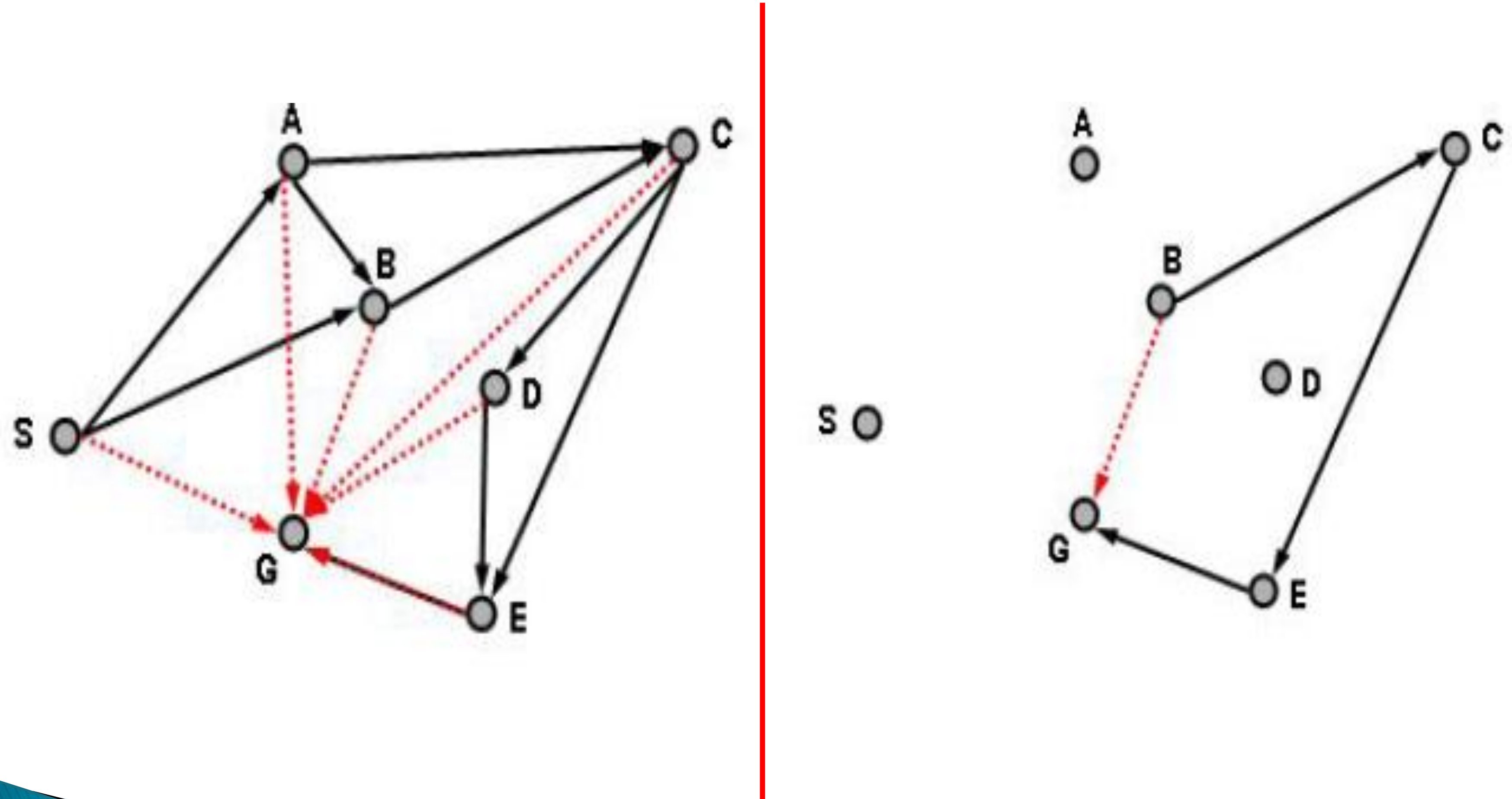
Heuristics

Admissible Heuristic

- An estimate that **always underestimates the real path length to *the goal*** is called **admissible estimate (heuristic).**
  - **Straight line distance** is an admissible estimate for path length in euclidian space.
- Uniform cost search is an instance of A*, If we set $h(n) = 0$.
  - Use of an admissible estimate gaurantee that Uniform-cost search will find the shortest path.

Uniform-cost search with admissible estimate (heuristic) is known as A* search.

Straight line Distance



Are all heuristics admissible?

- Given the **heuristic values** and **lenghts** in the Figure, are the heuristic values in the table addmissible?

A = OK

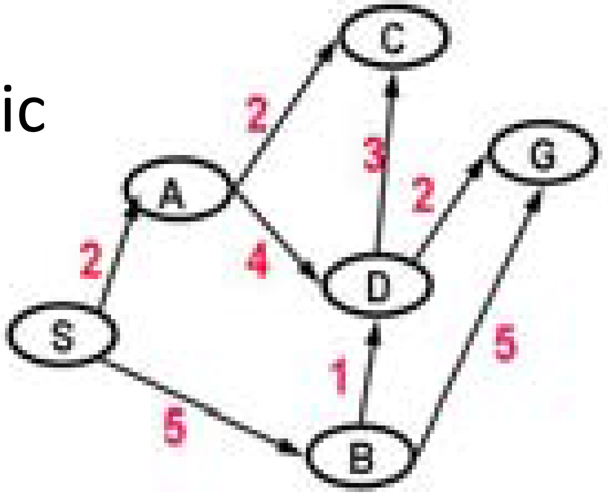
B = OK

C = OK

D = not OK, needs to be <= 2

S = not OK, should be 0.

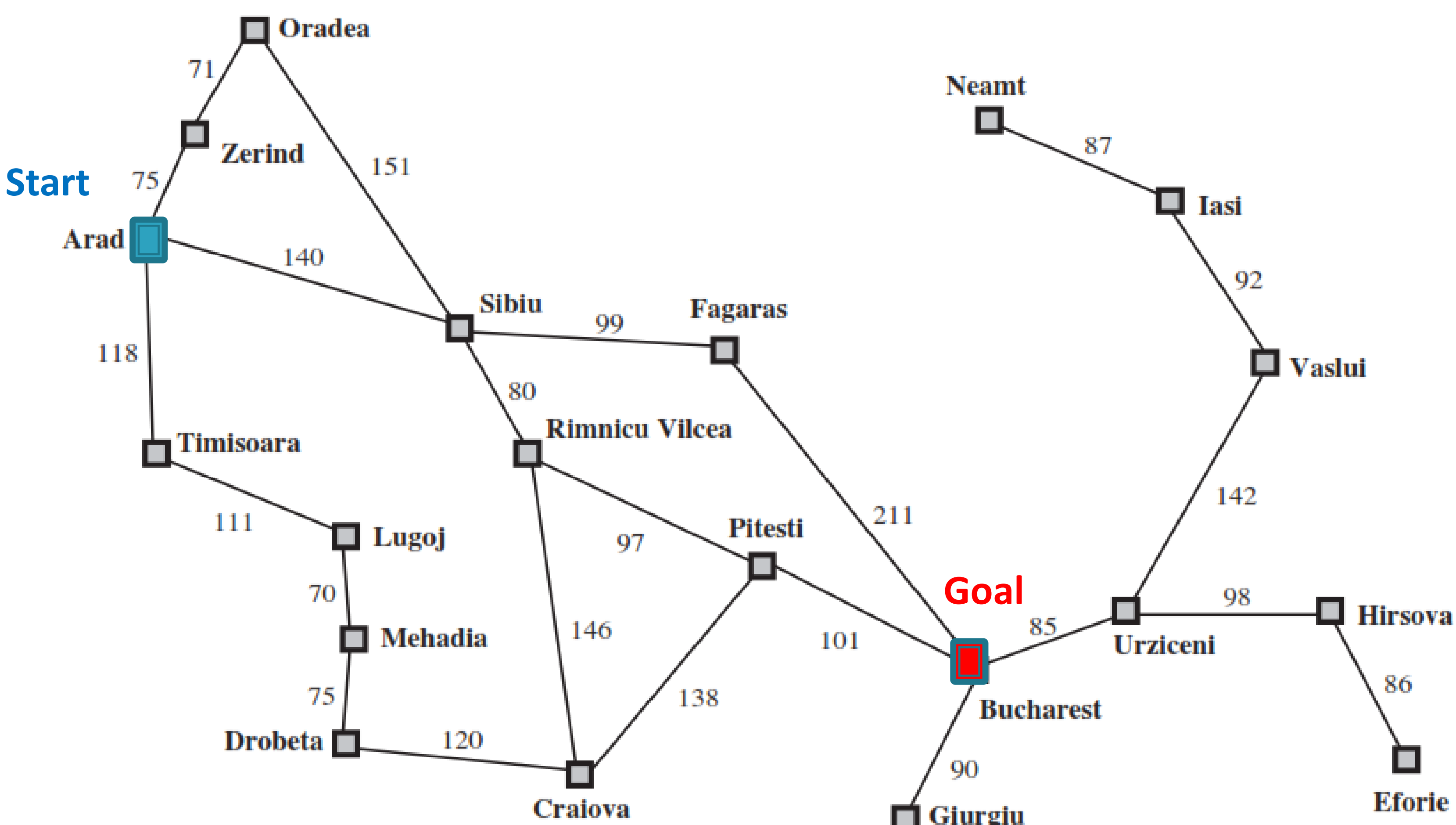
but the value of S would have no ill effect



Heuristic Values

A=2	C=1	S=10
B=3	D=4	G=0

Example



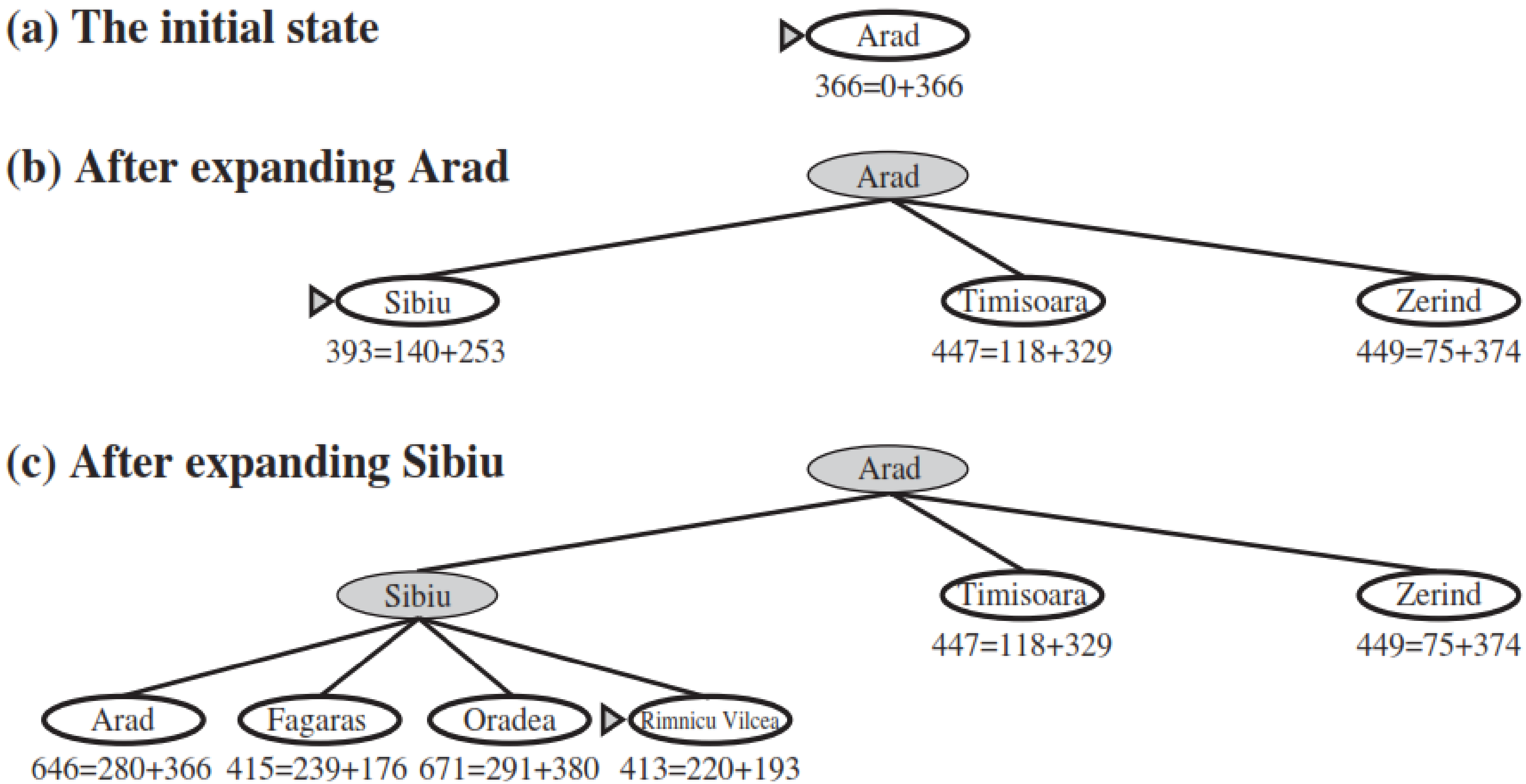
Oradea
71
Zerind
151
Start
75
Arad
140
Neamt
87
Iasi
92
Sibiu
99
Fagaras
118
80
Rimnicu Vilcea
Vaslui
Timisoara
111
Lugoj
97
Pitesti
211
142
70
Mehadia
146
101
Goal
85
Urziceni
98
Hirsova
75
138
Bucharest
86
Drobeta
120
Craiova
90
Giurgiu
Eforie

Example

Values of h_{SLD}—straight-line distances to Bucharest

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

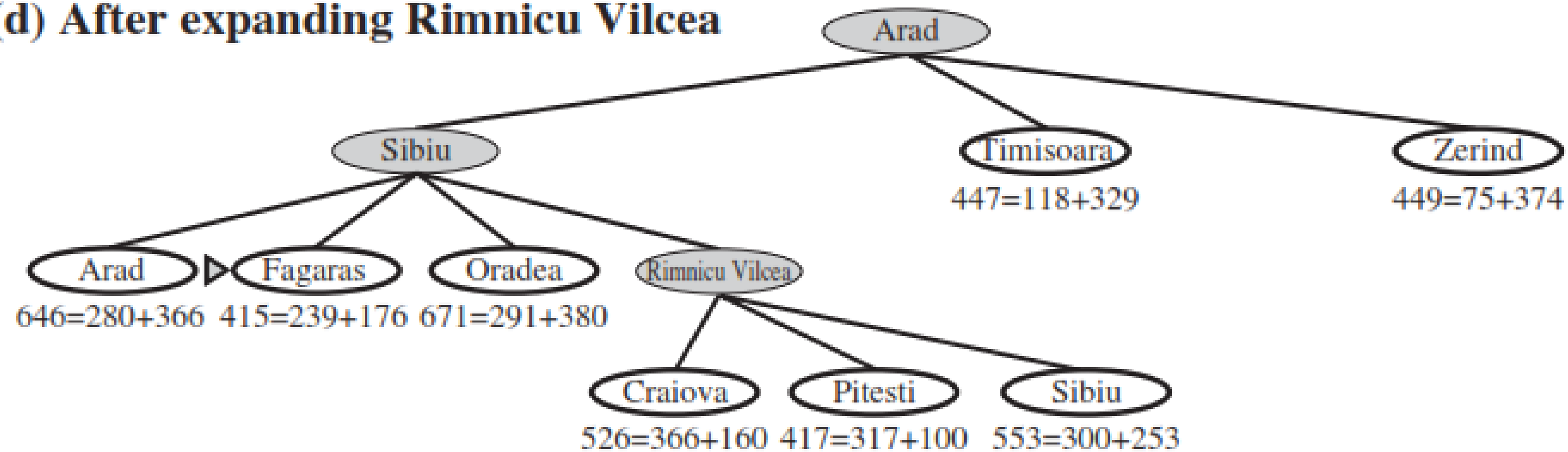
Example



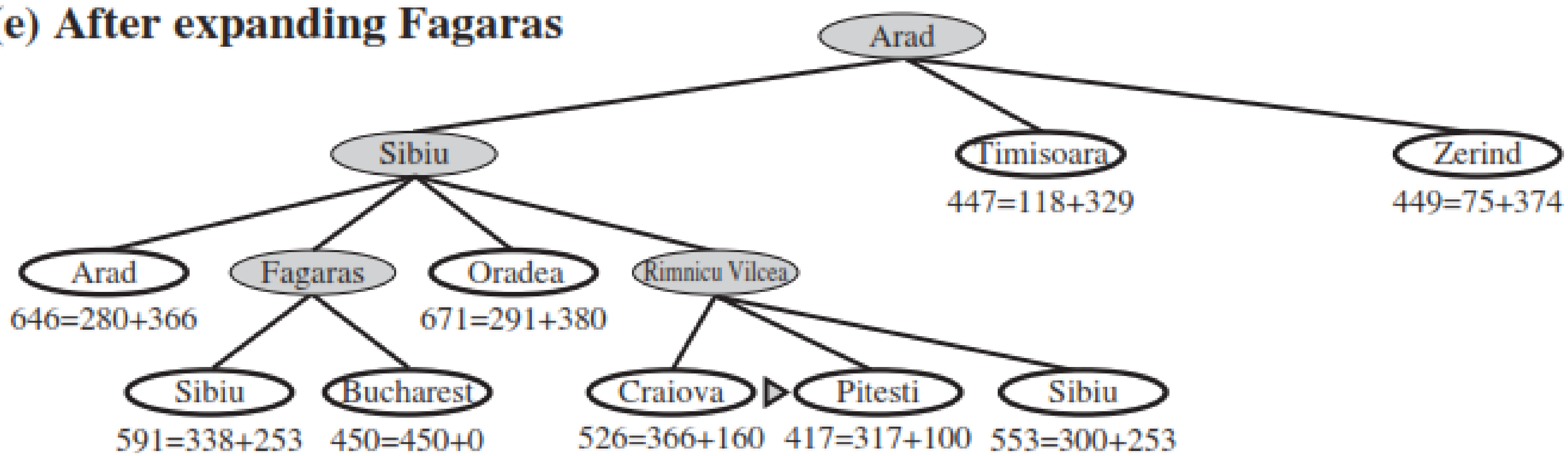
(a) The initial state
Arad
366=0+366
(b) After expanding Arad
Arad
Sibiu
393=140+253
Timisoara
447=118+329
Zerind
449=75+374
(c) After expanding Sibiu
Arad
Sibiu
Timisoara
447=118+329
Zerind
449=75+374
Arad
646=280+366
Fagaras
415=239+176
Oradea
671=291+380
Rimnicu Vilcea
413=220+193

Example

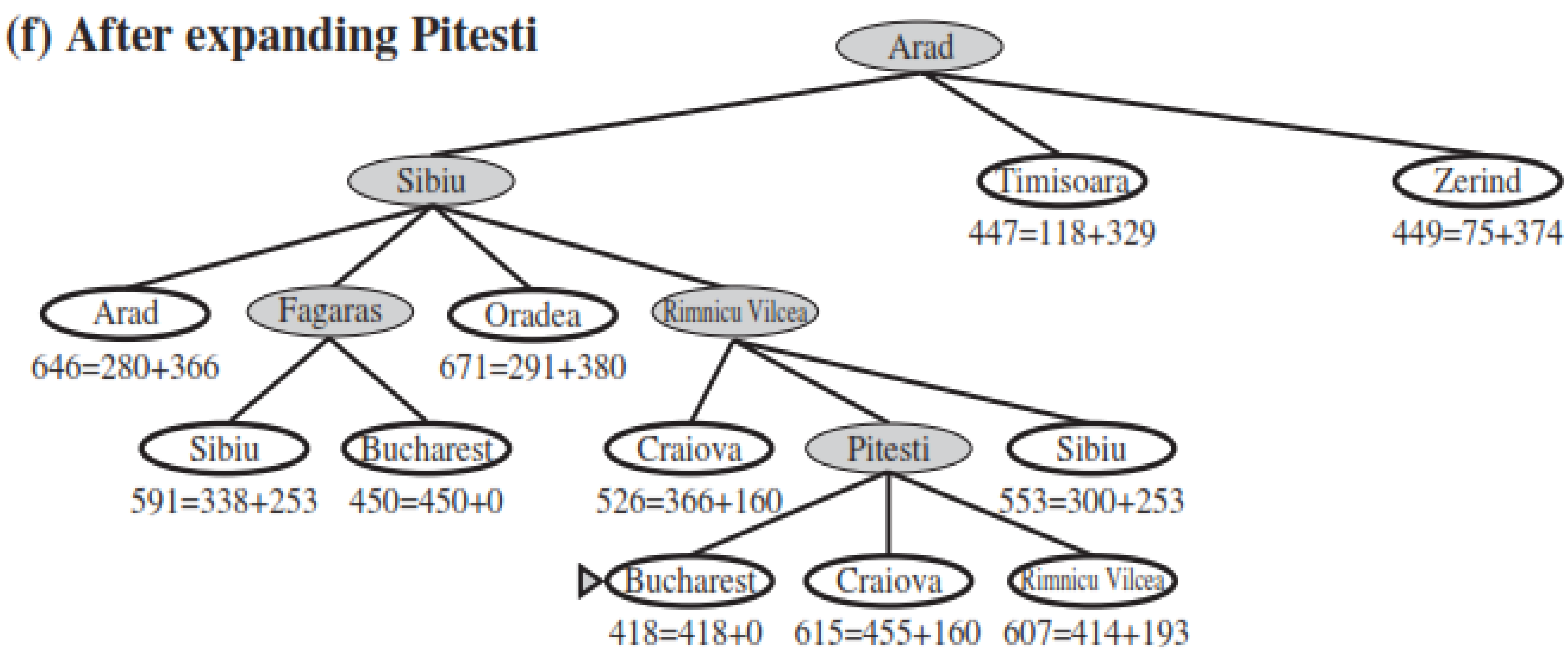
(d) After expanding Rimnicu Vilcea



(e) After expanding Fagaras



Example



A* search for Bucharest: Nodes are labeled with $f = g + h$. The h values are the straight-line distances to Bucharest.

Why use estimate of goal distance?

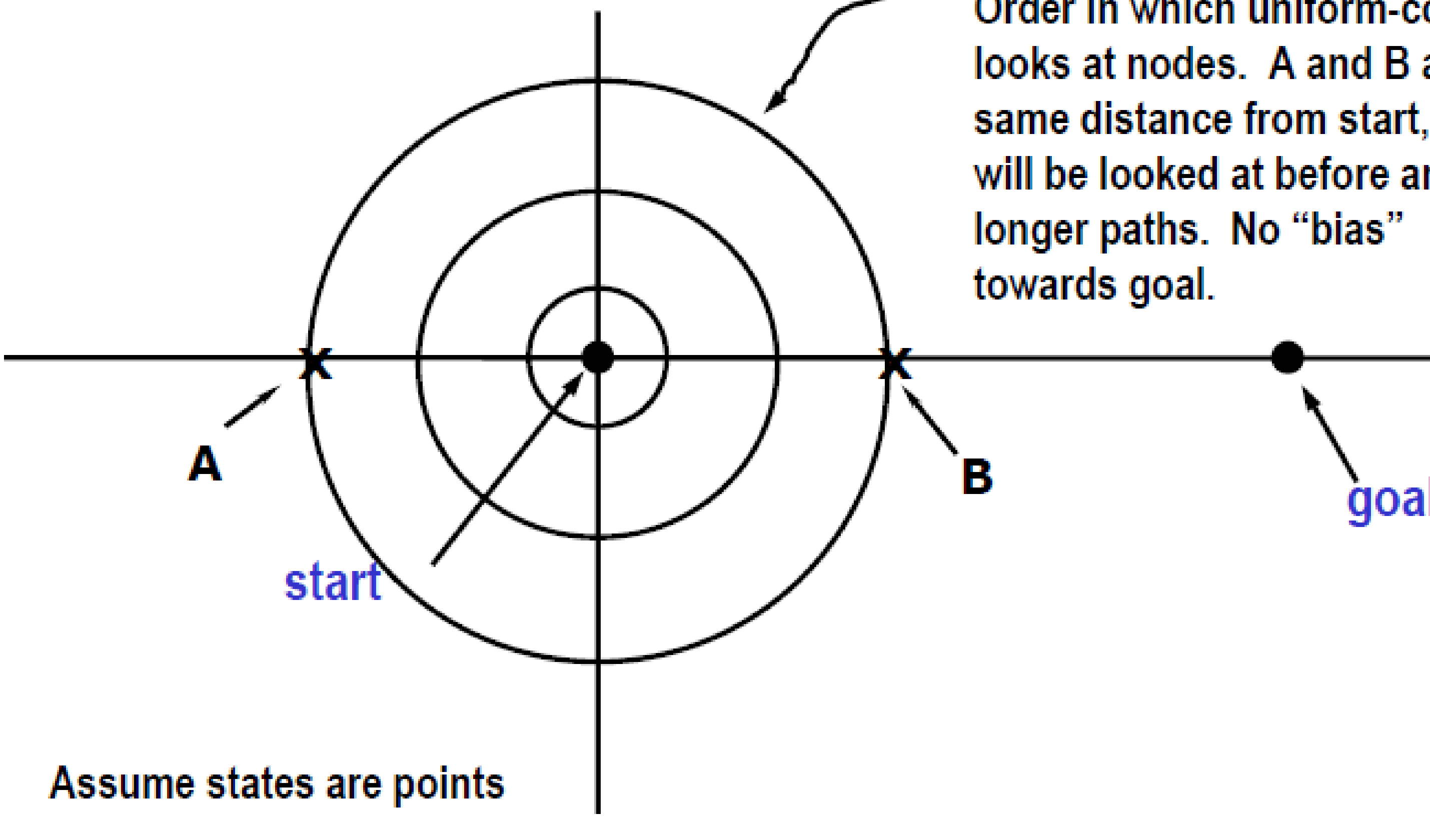
- You can think of A* as searching **contours** of distance from the start state + estimated distance to the goal.

- The **estimated/heuristic distance** term should **skew** the search in the direction of the goal.

- Heuristic doesn’t mislead.

- How do you find a heuristic?
  - In the path-planning problem, it wasn’t too hard to think of the shortest-line distance.

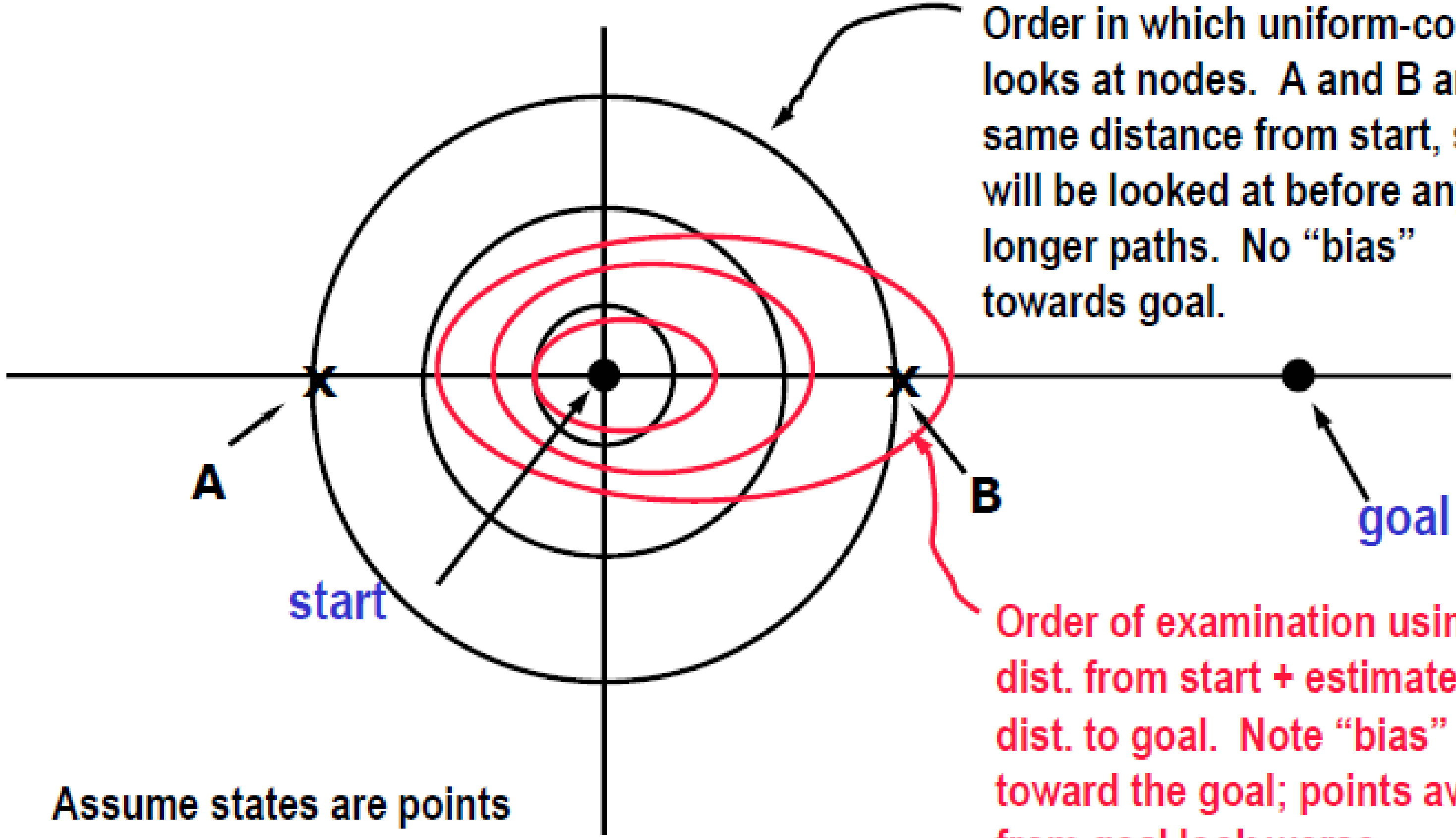
Why use estimate of goal distance?



Order in which uniform-cost looks at nodes. A and B are same distance from start, so will be looked at before any longer paths. No "bias" towards goal.

Assume states are points the Euclidean plane.

Why use estimate of goal distance?

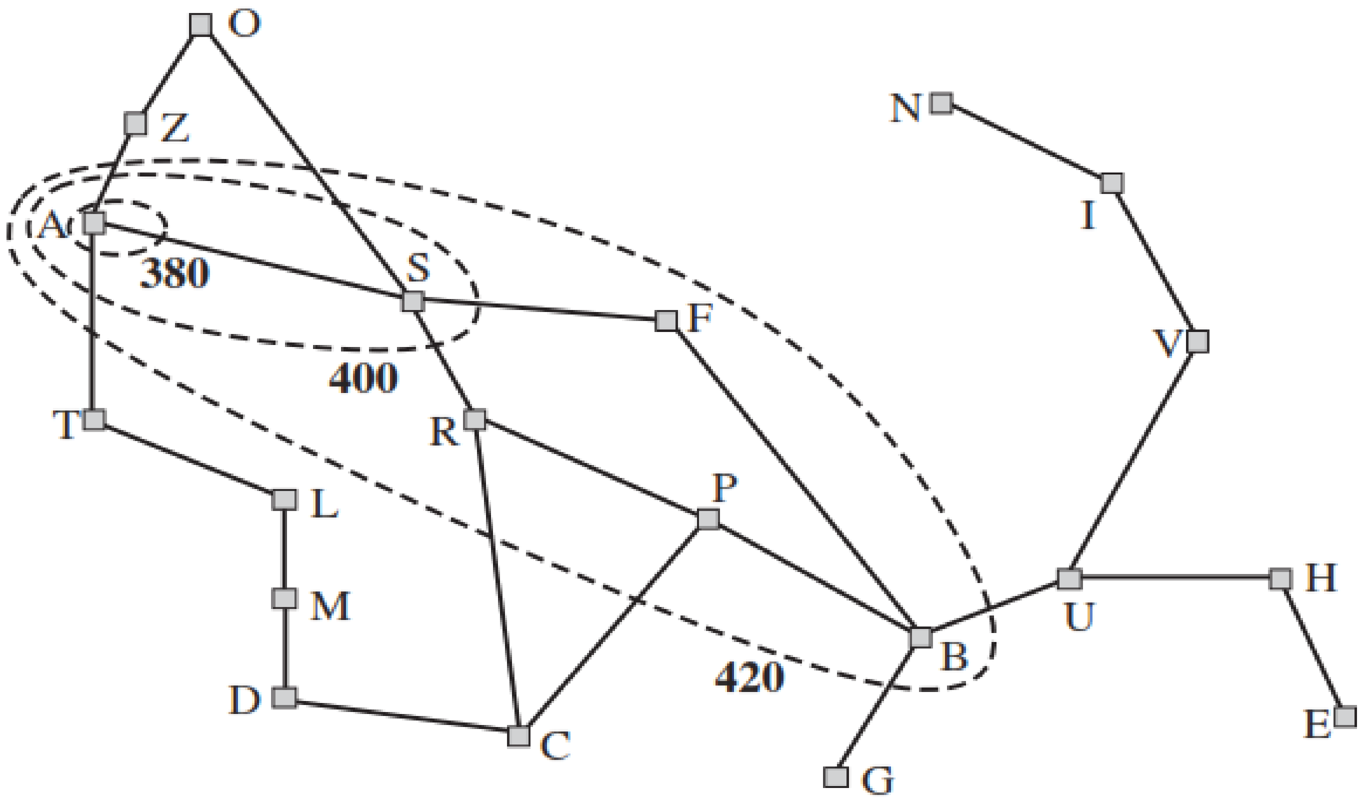


Order in which uniform-cost looks at nodes. A and B are same distance from start, so will be looked at before any longer paths. No "bias" towards goal.

Order of examination using dist. from start + estimate of dist. to goal. Note "bias" toward the goal; points away from goal look worse.

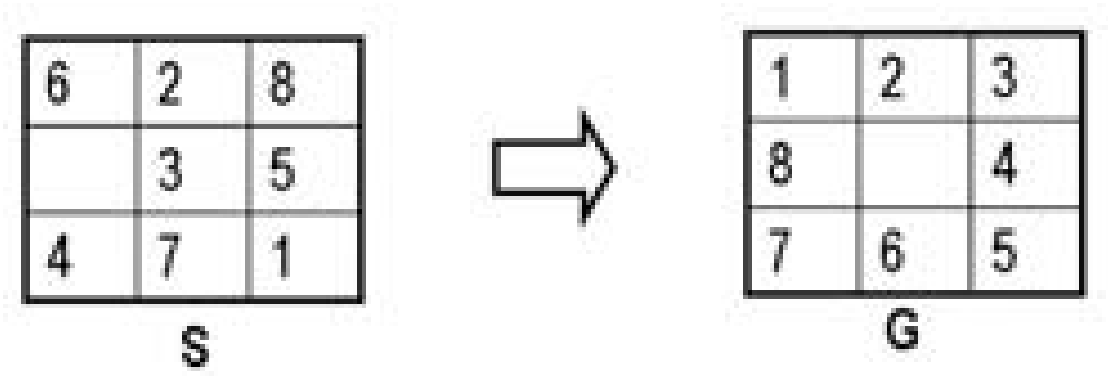
Assume states are points the Euclidean plane.

Why use estimate of goal distance?



Map of Romania showing contours at $f = 380$, $f = 400$, and $f = 420$, with Arad as the start state.

Admissible Heuristic ...8-puzzle



- **The goal** is to arrange the pieces as in the goal state on the right.
- **A move** in this game is as sliding the **"empty"** space to one of its **nearest vertical or horizontal neighbours.**
- Move tiles to reach goal

Admissible Heuristic ...8-puzzle

6	2	8
	3	5
4	7	1

S

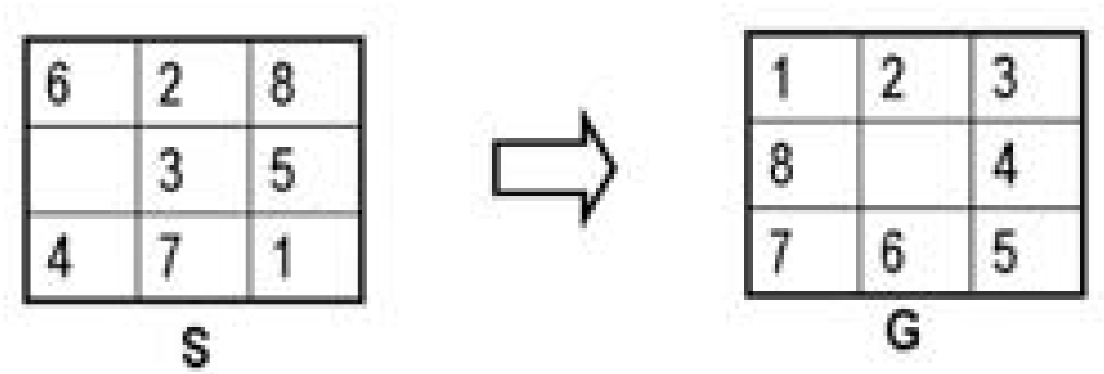
1	2	3
8		4
7	6	5

G

- Each move can at best decrease by one, the **"Manhattan distance"** of a tile from its goal.
- Manhattan distance, the metric of the Euclidean plane, is defined as

$$m\big((x_1,y_1)(x_2,y_2)\big) = |x_1 - x_2| + |y_1 - y_2|$$

Admissible Heuristic ...8-puzzle



The alternative underestimates of "distance" (the number of moves) to goal:

- Number of misplaced tiles (7 in this case)
- Sum of manhattan distance, which is 17
  - **tile 1 = 4, tile 2 = 0, tile 3 = 2, tile 4 = 3, tile 5 = 1, tile 6 = 3, tile 7 = 1, tile 8 = 3**

Admissible Heuristics ... Dominance

- If $\boldsymbol{h_2(n) \geq h_1(n)}$ for all n (both admissible) then h_2 **dominates** h_1 and is better for search

$$
\begin{aligned}
d = 14 \quad & \text{IDS} = 3{,}473{,}941 \text{ nodes} \\
& \text{A}^*(h_1) = 539 \text{ nodes} \\
& \text{A}^*(h_2) = 113 \text{ nodes} \\
d = 24 \quad & \text{IDS} \approx 54{,}000{,}000{,}000 \text{ nodes} \\
& \text{A}^*(h_1) = 39{,}135 \text{ nodes} \\
& \text{A}^*(h_2) = 1{,}641 \text{ nodes}
\end{aligned}
$$

- Given admissible heuristics $h_1 \ldots h_b$,

$$\boldsymbol{h(n) = max[h_1(n) \ldots h_b(n)]}$$

is also admissible and dominates the other heuristics.

Admissible Heuristics … Dominance

$$h_2(n) \geq h_1(n)$$

	Search Cost (nodes generated)			Effective Branching Factor		
d	IDS	$A^*(h_1)$	$A^*(h_2)$	IDS	$A^*(h_1)$	$A^*(h_2)$
2	10	6	6	2.45	1.79	1.79
4	112	13	12	2.87	1.48	1.45
6	680	20	18	2.73	1.34	1.30
8	6384	39	25	2.80	1.33	1.24
10	47127	93	39	2.79	1.38	1.22
12	3644035	227	73	2.78	1.42	1.24
14	–	539	113	–	1.44	1.23
16	–	1301	211	–	1.45	1.25
18	–	3056	363	–	1.46	1.26
20	–	7276	676	–	1.47	1.27
22	–	18094	1219	–	1.48	1.28
24	–	39135	1641	–	1.48	1.26

Consistency

Consistency

To implement A* Search, **heurstic h needs to be consistant** (sometimes monotonicity):

- The goal states have a heuristic estimate of zero.

$$h(n_i) = 0 \qquad if\ node\ n_i\ is\ the\ goal$$

- *The difference in the heuristic estimate between one state and its descendant must be less than or equal to the actual path cost.*

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

$$h(n) - h(n') \le c(n, a, n')$$

$$where\ n'\ is\ the\ successor\ of\ n.$$

Consistency Lemma

- ***If $h(n)$ is consistent, then the values of $f(n)$ along any path are nondecreasing.***

$$f(n') \geq f(n)$$

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

- ***Proof:***

$$Suppose\ n'\ is\ the\ successor\ of\ n$$

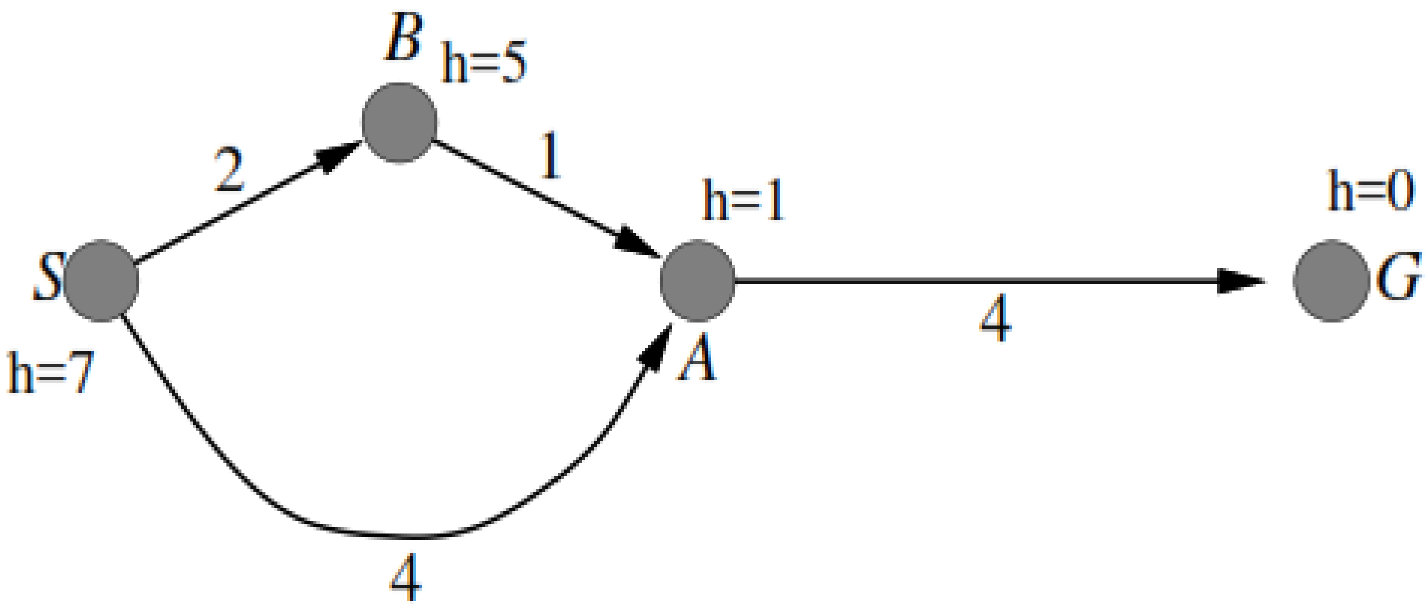
$$g(n') = g(n) + c(n, a, n')$$

$$f(n') = g(n') + h(n')$$
$$f(n') = g(n) + c(n, a, n') + h(n')$$
$$\geq g(n) + h(n)$$
$$f(n') \geq f(n)$$

Example

Example

- Are all heuristics admissible and consistent in this graph?



Reading Material

- **Artificial Intelligence,** A Modern Approach

  Stuart J. Russell and Peter Norvig

  - **Chapter 3.**