Artificial Intelligence (CS-2002)

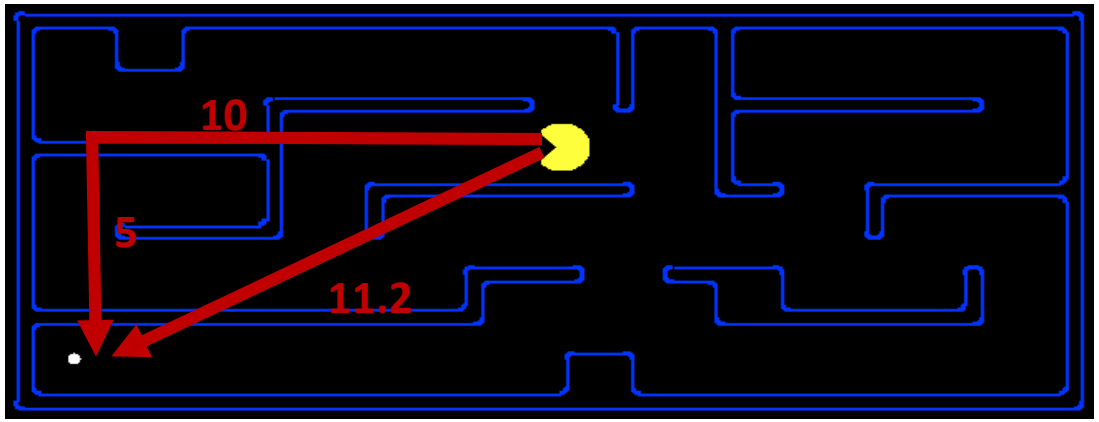
Informed Search Strategies

Objectives

- Heuristics
- Greedy Best first Search
- A*

Search Heuristics

- The algorithms have additional information along with the goal state, which helps in more efficient searching. This information is obtained by something called a *heuristic.*
- A heuristic is:
  - A function that *estimates* how close a state is to a goal
  - Designed for a particular search problem
  - Examples: Manhattan distance, Euclidean distance for pathing



Greedy Best First Search

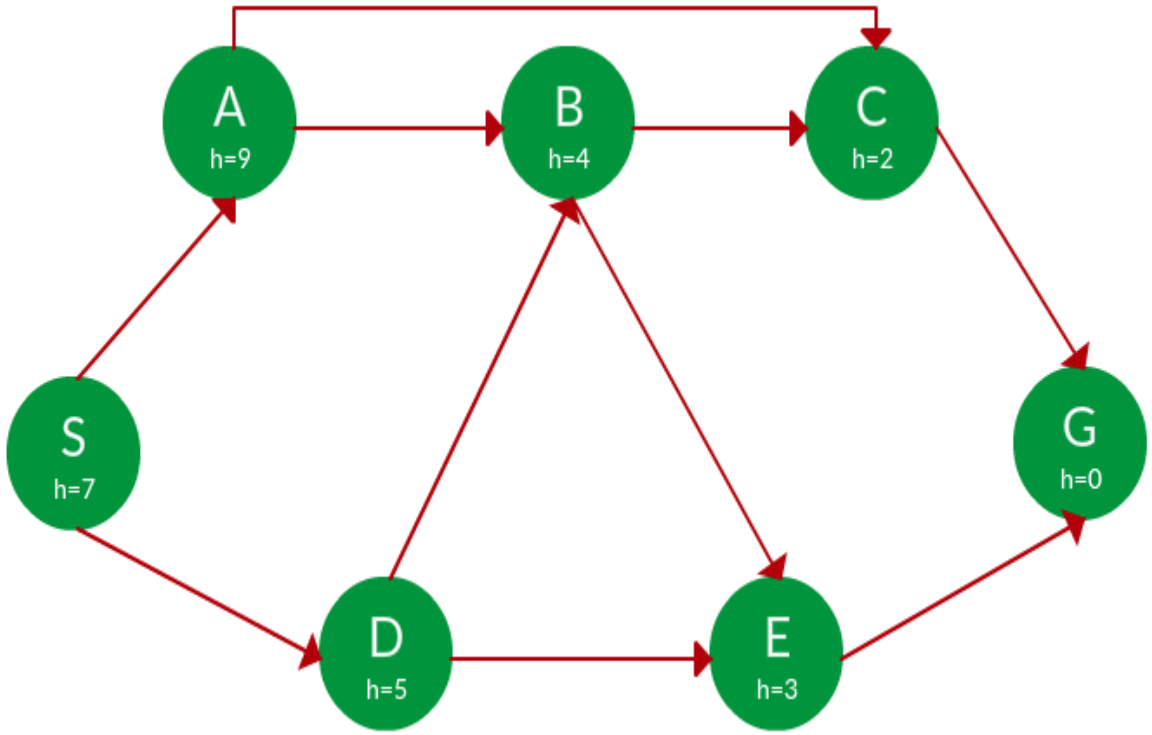
In greedy search, we expand the node closest to the goal node. The “closeness” is estimated by a heuristic `h(x)` .

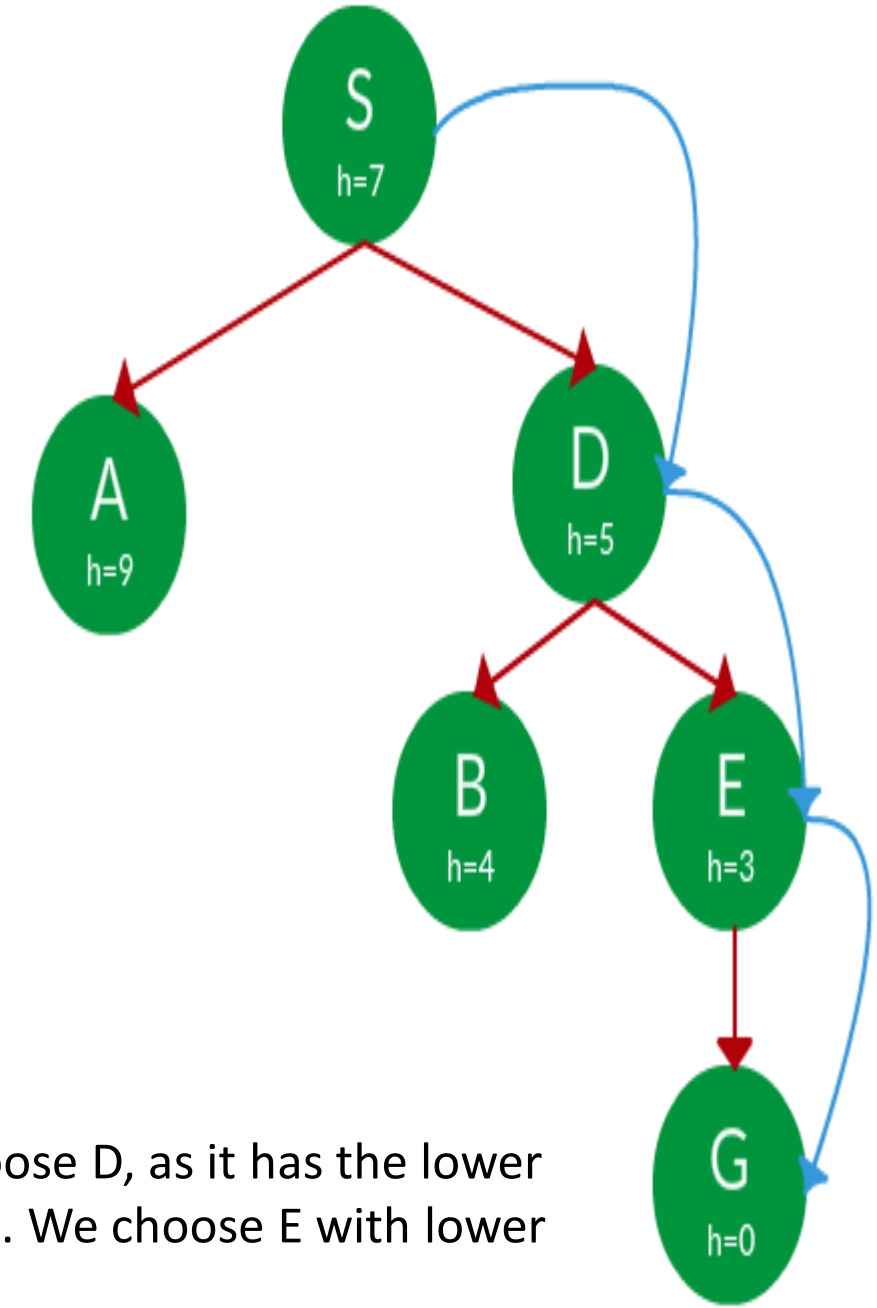
Heuristic: A heuristic `h` is defined as-
`h(x)` = Estimate of distance of node `x` from the goal node.
Lower the value of `h(x)`, closer is the node from the goal.

Strategy: Expand the node closest to the goal state, *i.e.* expand the node with lower `h` value.

Example:

Find the path from S to G using greedy search.



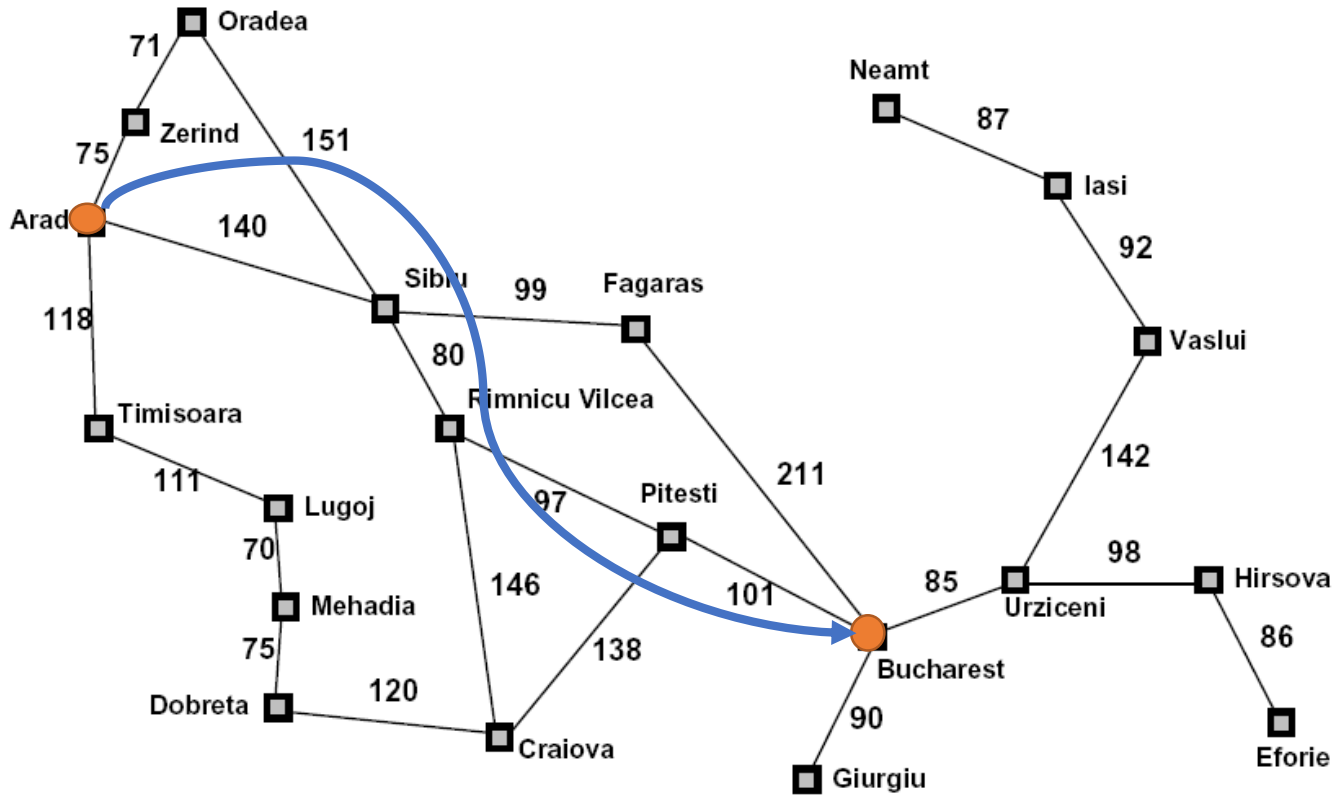


Solution

Starting from S, we can traverse to A(h=9) or D(h=5). We choose D, as it has the lower heuristic cost. Now from D, we can move to B(h=4) or E(h=3). We choose E with lower heuristic cost. Finally, from E, we go to G(h=0).

Path: S -> D -> E -> G

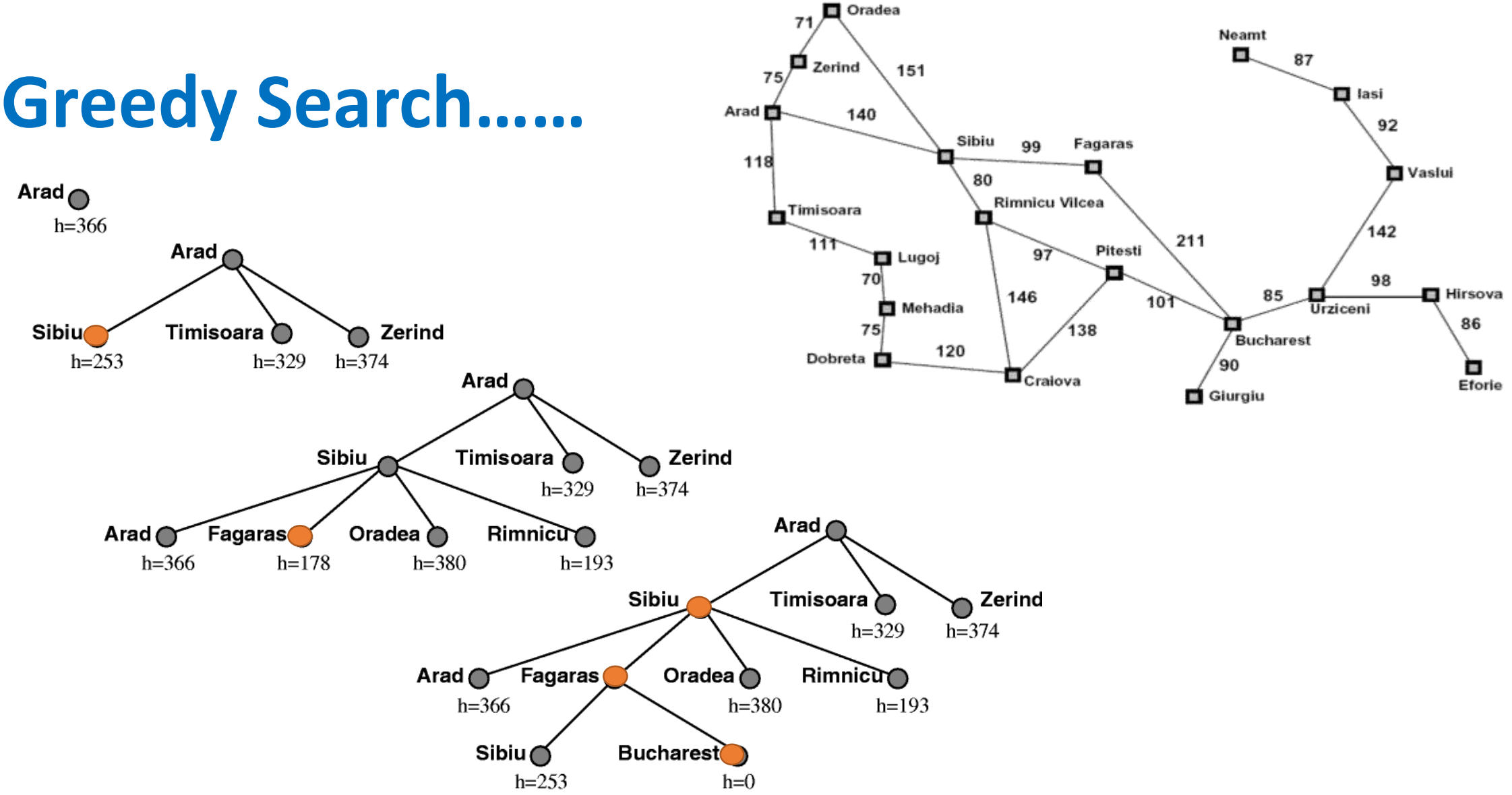
Travel Sales Man Problem using Greedy Search



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

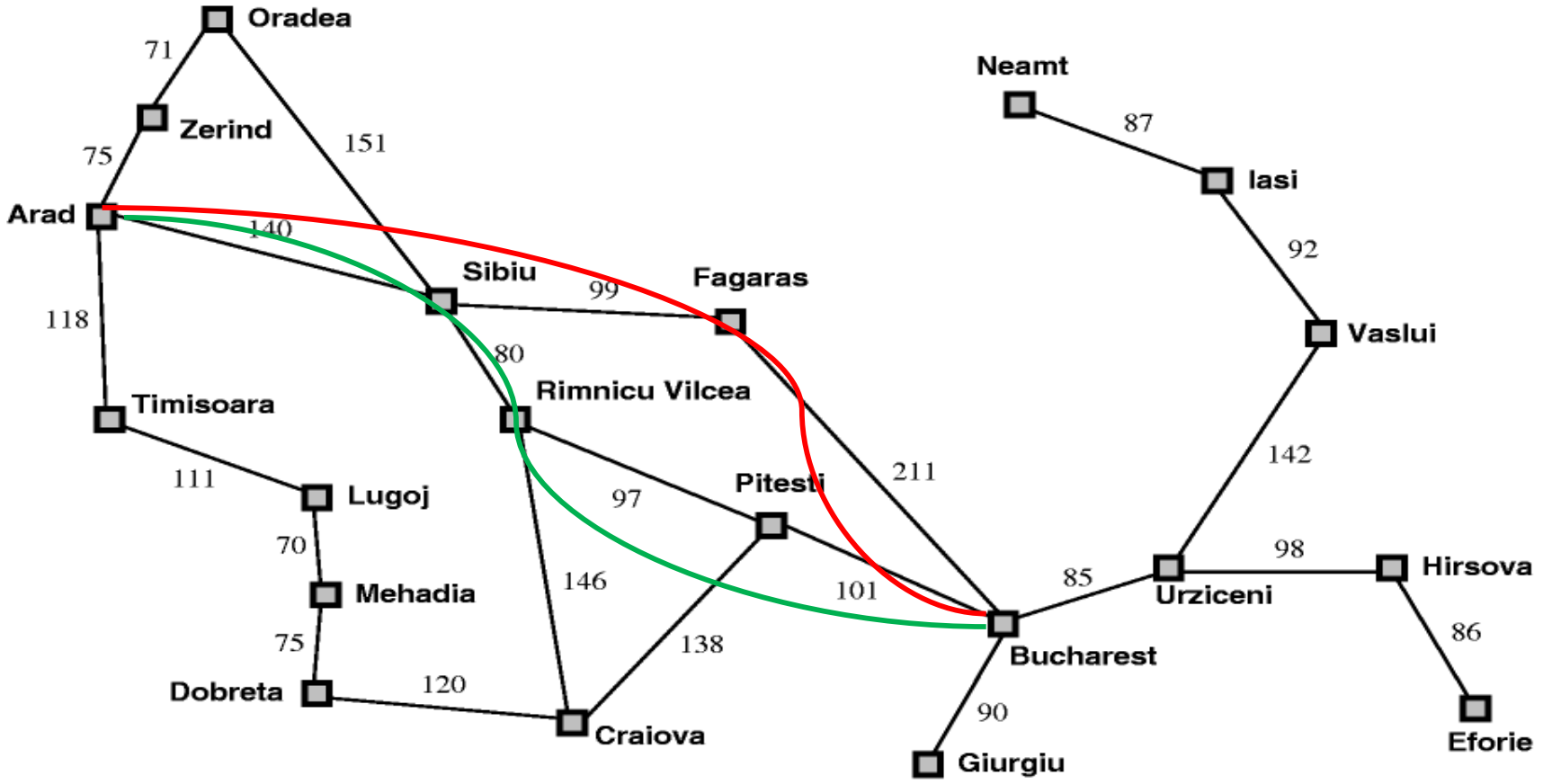
h(x)

Greedy Search……



What can go wrong?

Not optimal



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

Red Path (GBF)=431

Green Path (UCS)= 418

Greedy search leads to a minimal cost search because no node off the solution path is expanded. However, it does not find the optimal path: the path it found via Sibiu and Fagaras to Bucharest is 13 miles longer than the path through Pimnicu Vilcea and Pitesti.

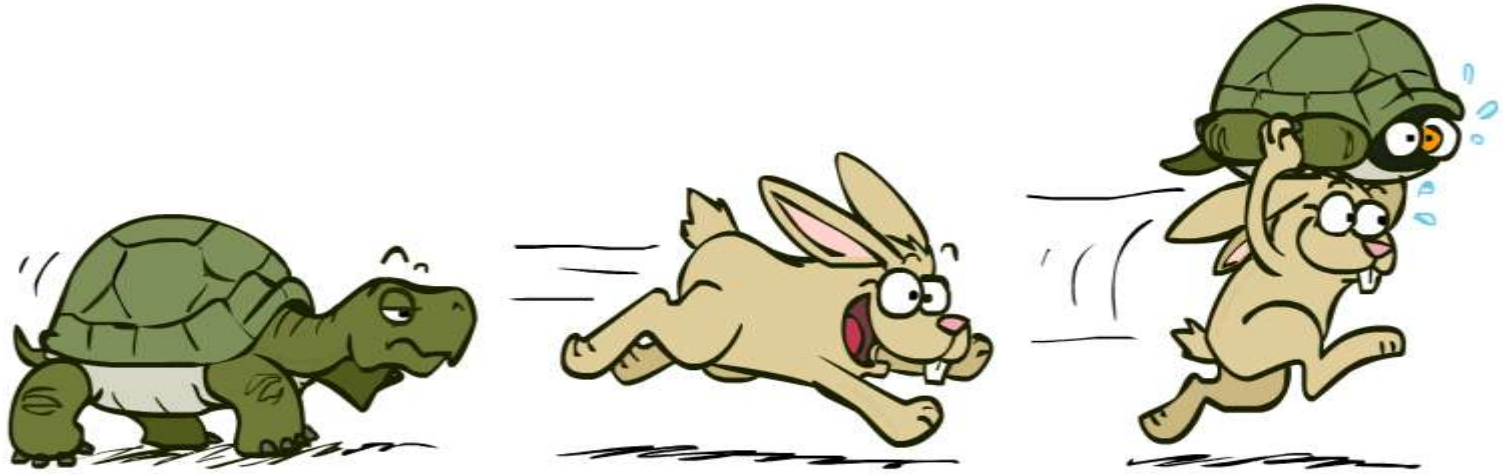
A* Search

- A* Search, combines the strengths of uniform-cost search and greedy search. In this search, the *heuristic is the summation of the cost in UCS*, denoted by g(x), and the **cost in greedy search**, denoted by h(x). The *summed cost* is denoted by f(x).

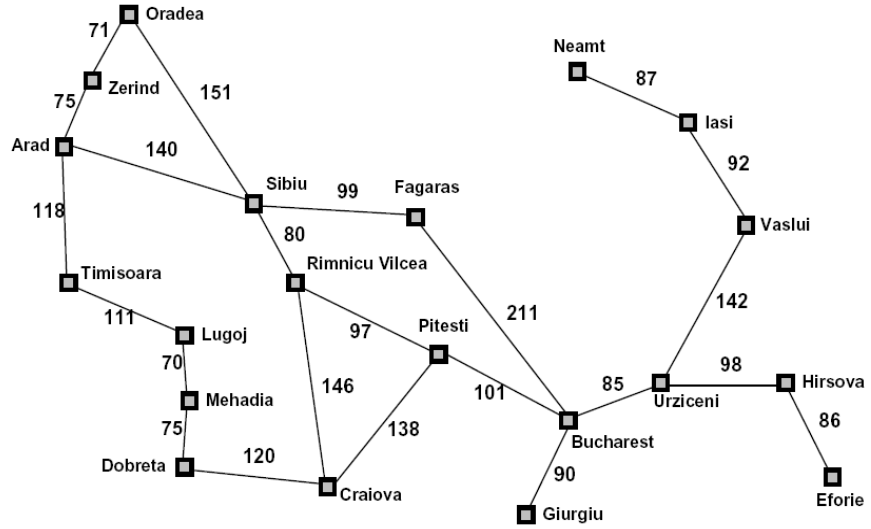
- **Heuristic:** The following points should be noted w.r.t heuristics in A* search.

- A* Search orders by the sum: **F(N) = G(N) + H(N)**

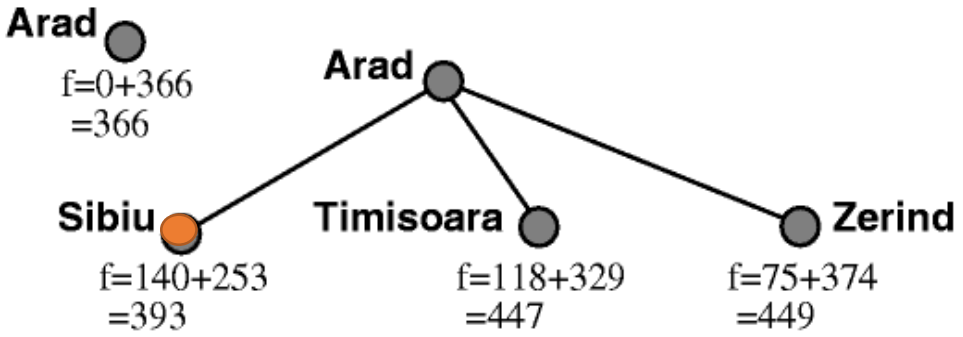
Uniform-cost orders by path cost,
or *backward cost* g(n)
Greedy orders by goal proximity,
or *forward cost* h(n)

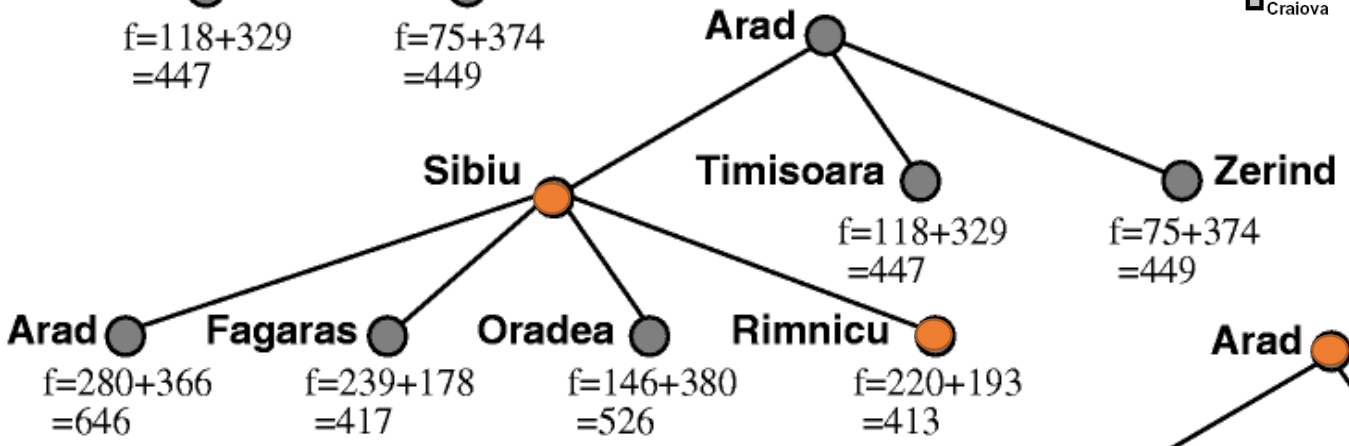


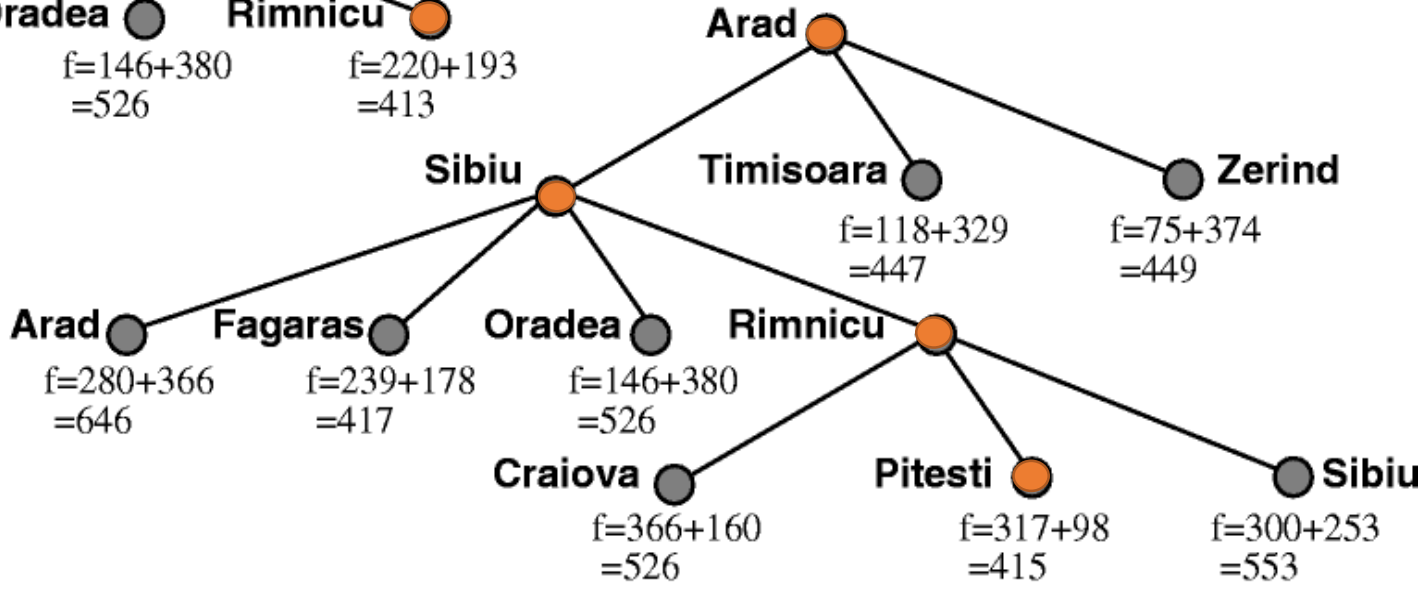
A* Search



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







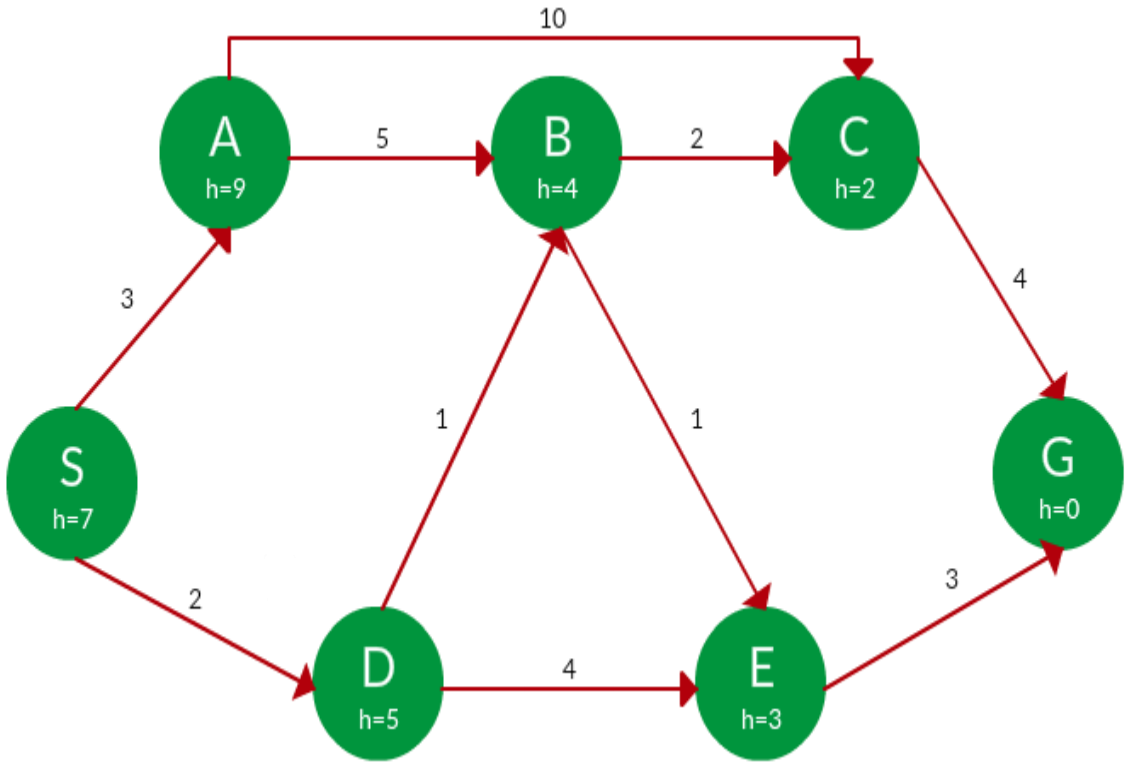
Admissible Heuristics

- A heuristic h is *admissible* (optimistic) if:

$$0 \leq h(n) \leq h^*(n)$$

where $h^*(n)$ is the true cost to a nearest goal

A* Search



$$0 \leq h(n) \leq h^*(n)$$

Path: S -> D -> B -> E -> G
Cost: 7

PATH	H(X)	G(X)	F(X)
S	7	0	7
S -> A	9	3	12
S -> D ✓	5	2	7
S -> D -> B ✓	4	2 + 1 = 3	7
S -> D -> E	3	2 + 4 = 6	9
S -> D -> B -> C ✓	2	3 + 2 = 5	7
S -> D -> B -> E ✓	3	3 + 1 = 4	7
S -> D -> B -> C -> G	0	5 + 4 = 9	9
S -> D -> B -> E -> G ✓	0	4 + 3 = 7	7

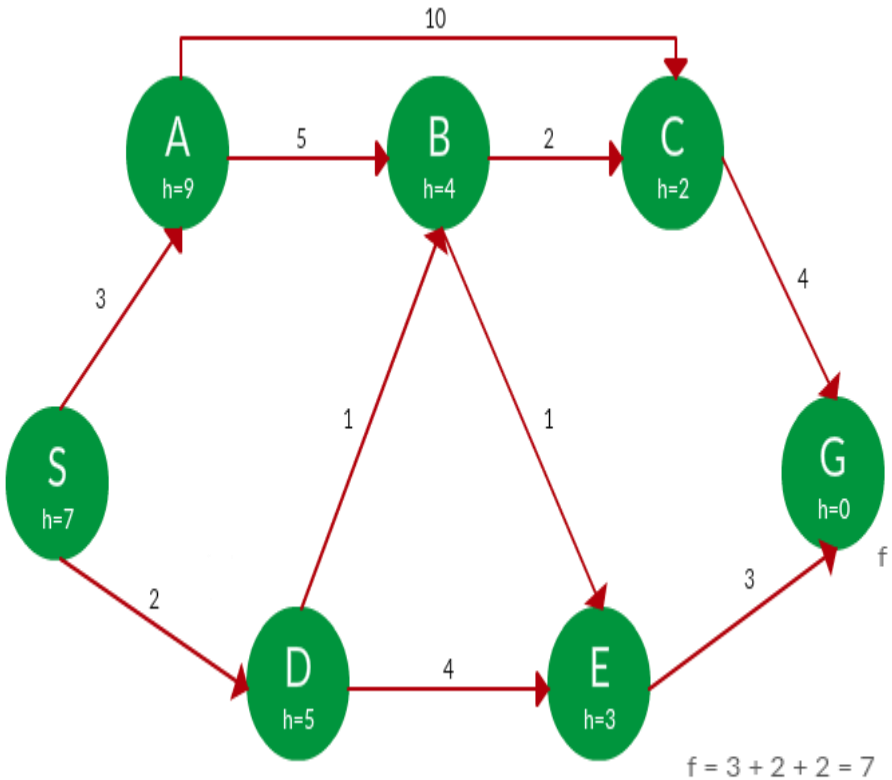
Consistency of Heuristics

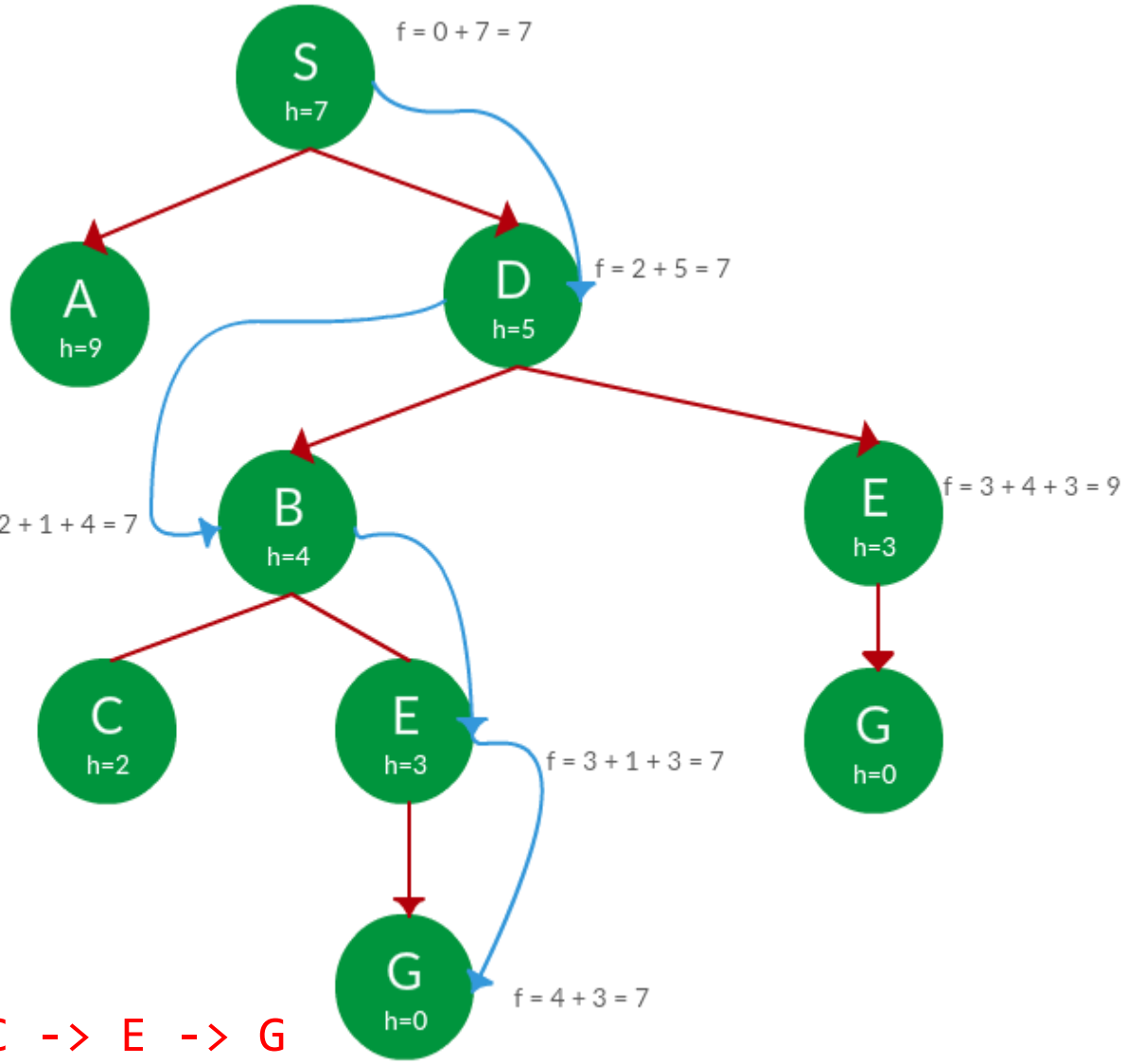
- Main idea: estimated heuristic costs ≤ actual costs
  - Admissibility: heuristic cost ≤ actual cost to goal

    $h(A) \leq$ actual cost from A to G
  - Consistency: heuristic “arc” cost ≤ actual cost for each arc

    $h(A) - h(C) \leq$ cost(A to C)

A* Search





Explored Nodes
S
D
B
E
G

Path: S -> D -> B -> C -> E -> G

Cost: 7