innovate achieve lead

BITS Pilani
Pilani Campus

Session 5: Informed Search (A*)
And its proof of optimiality

Session Content

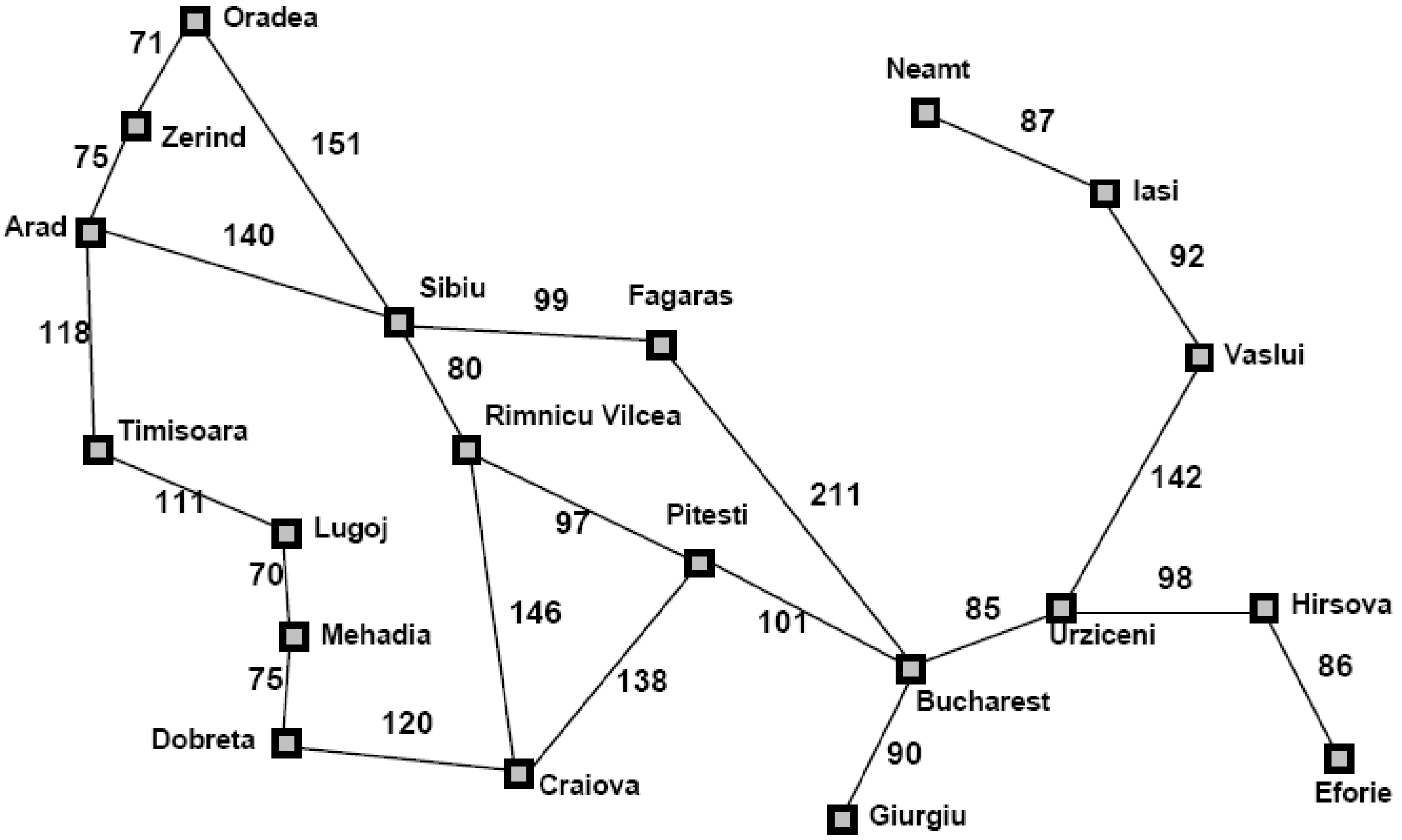
- Informed Search (A*)
- Admissibility of heuristic
- Optimality of A*

A* Search

- Idea: avoid expanding paths that are already expensive
- Evaluation function $f(n) = g(n) + h(n)$

- $g(n)$ = cost so far to reach n
- $h(n)$ = estimated cost from n to goal
- $f(n)$ = estimated total cost of path through n to goal

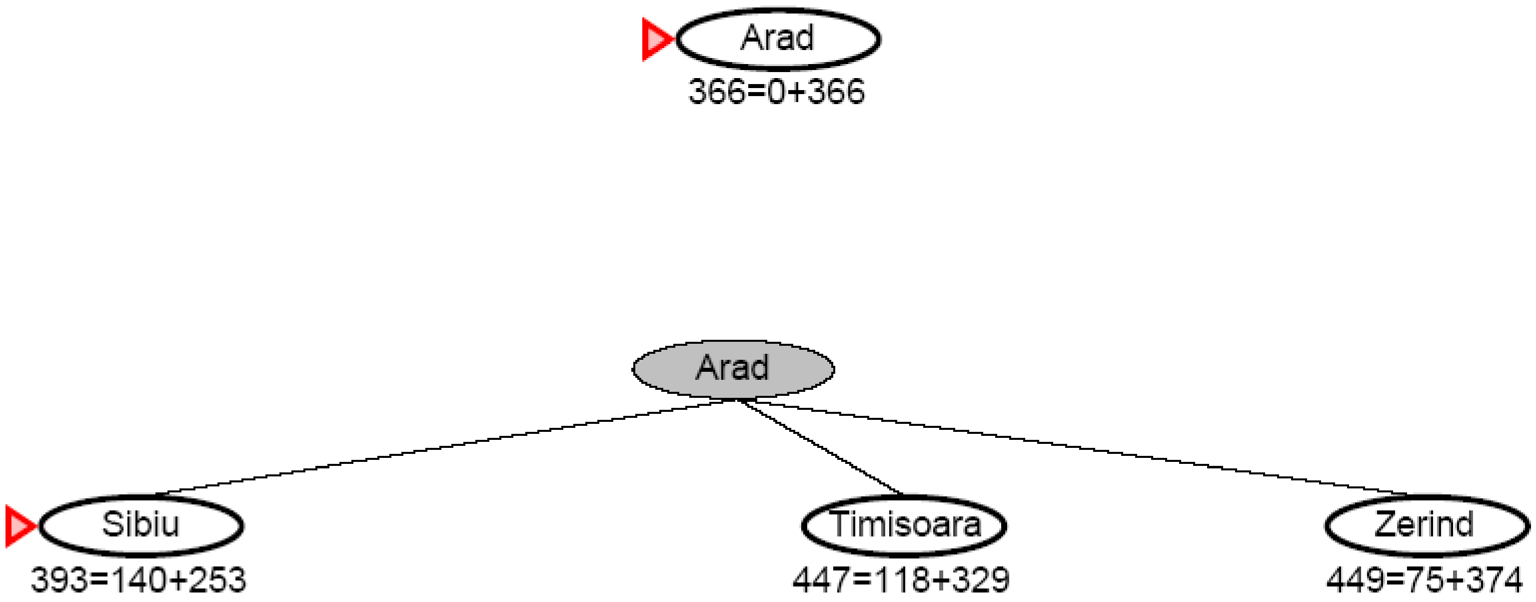
A* Search



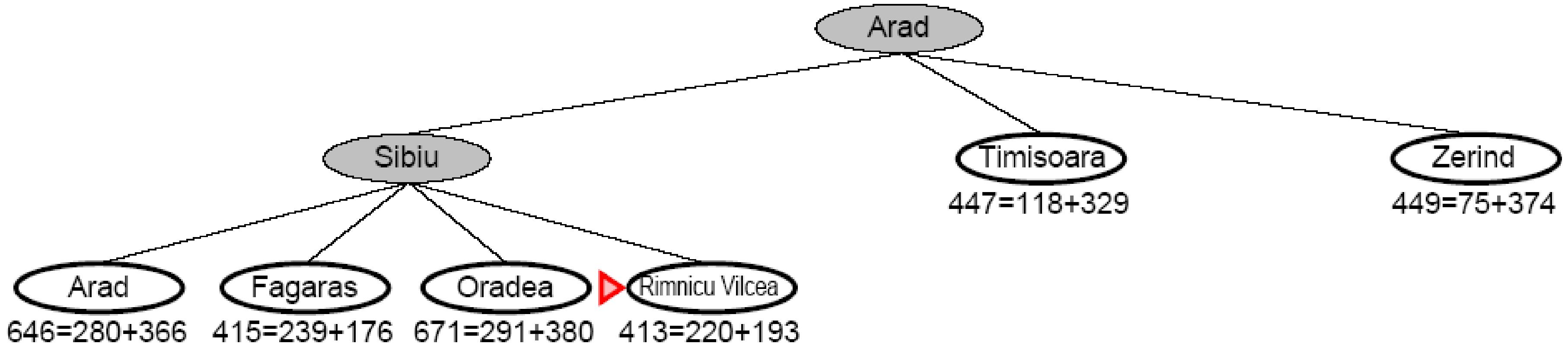
Straight−line distance to Bucharest

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

A* Search

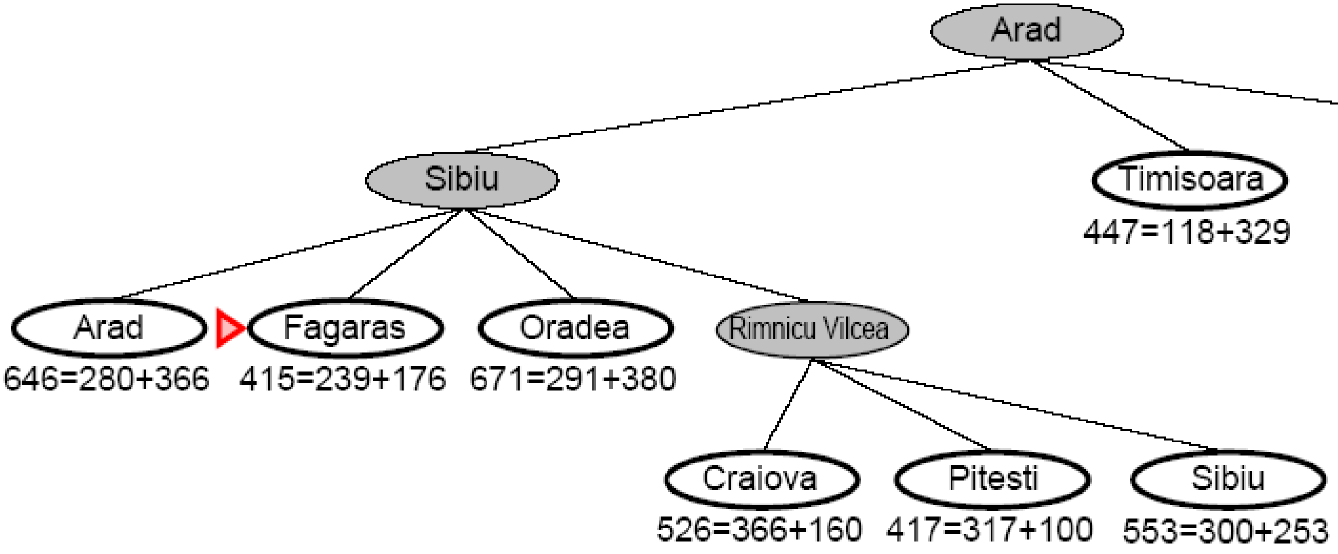

Arad
366=0+366
Arad
Sibiu
393=140+253
Timisoara
447=118+329
Zerind
449=75+374

A* Search



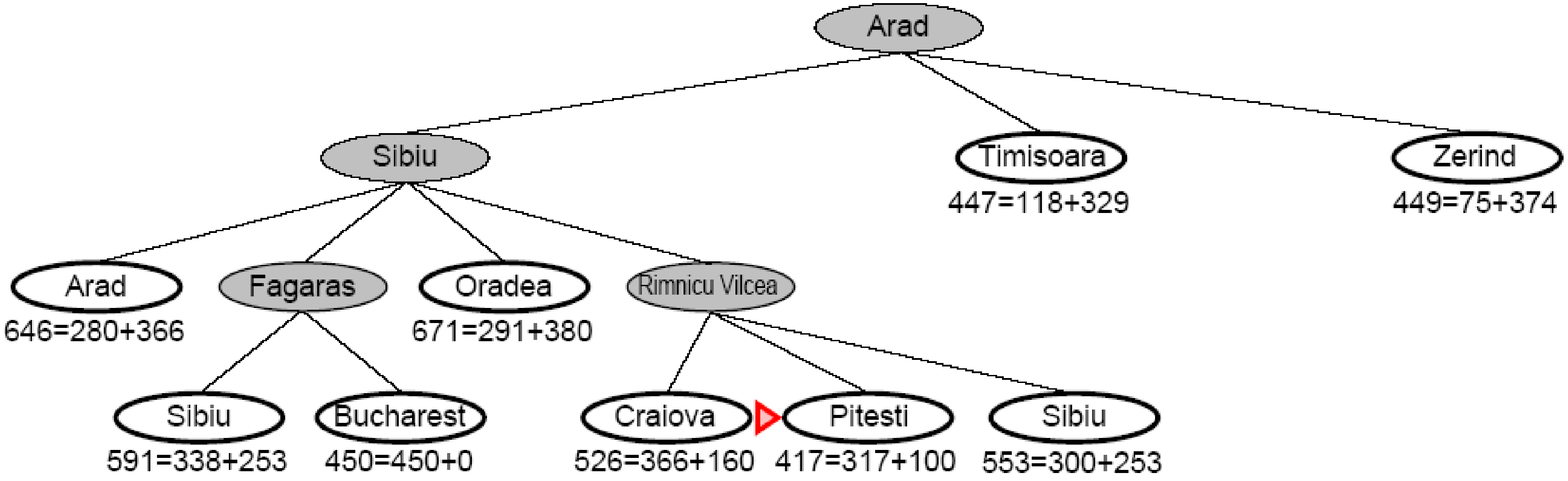
Arad
Sibiu
Timisoara
447=118+329
Zerind
449=75+374
Arad
646=280+366
Fagaras
415=239+176
Oradea
671=291+380
Rimnicu Vilcea
413=220+193

A* Search

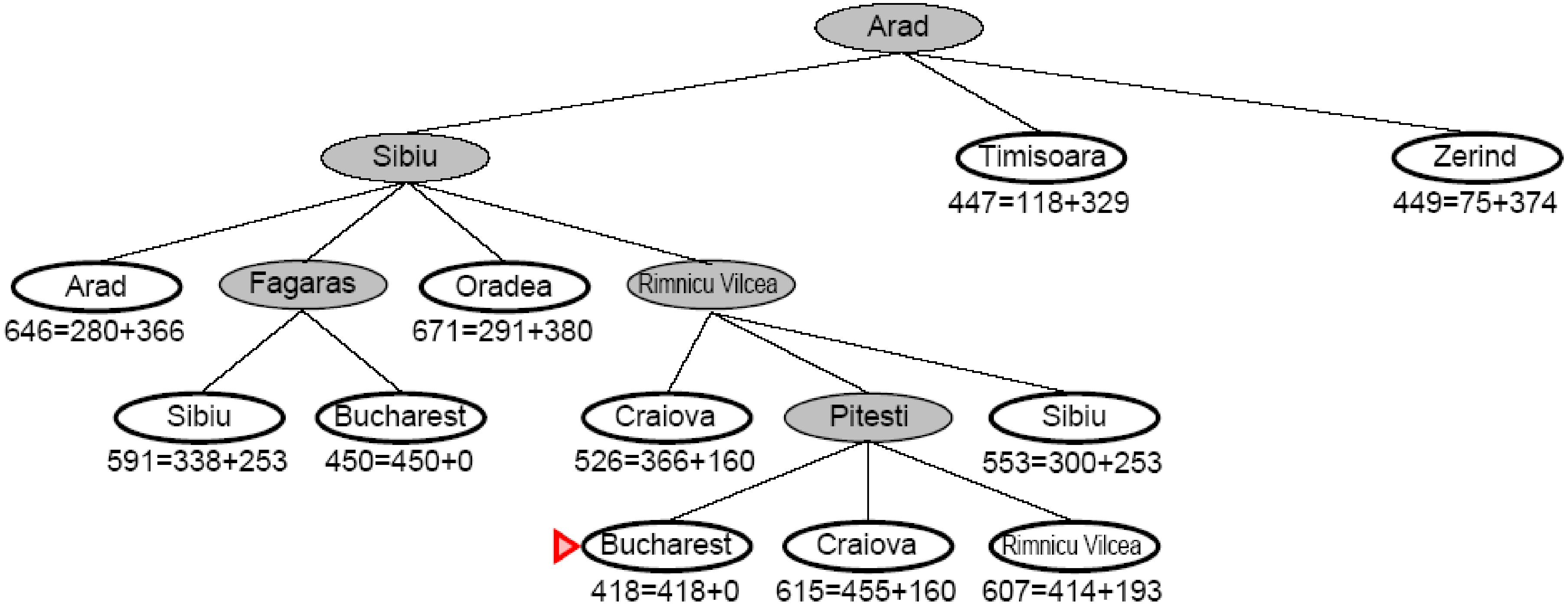


Arad
Sibiu
Timisoara
447=118+329
Zerind
449=75+374
Arad
646=280+366
Fagaras
415=239+176
Oradea
671=291+380
Rimnicu Vilcea
Craiova
526=366+160
Pitesti
417=317+100
Sibiu
553=300+253

A* Search


Arad
Sibiu
Timisoara
447=118+329
Zerind
449=75+374
Arad
646=280+366
Fagaras
Oradea
671=291+380
Rimnicu Vilcea
Sibiu
591=338+253
Bucharest
450=450+0
Craiova
526=366+160
Pitesti
417=317+100
Sibiu
553=300+253

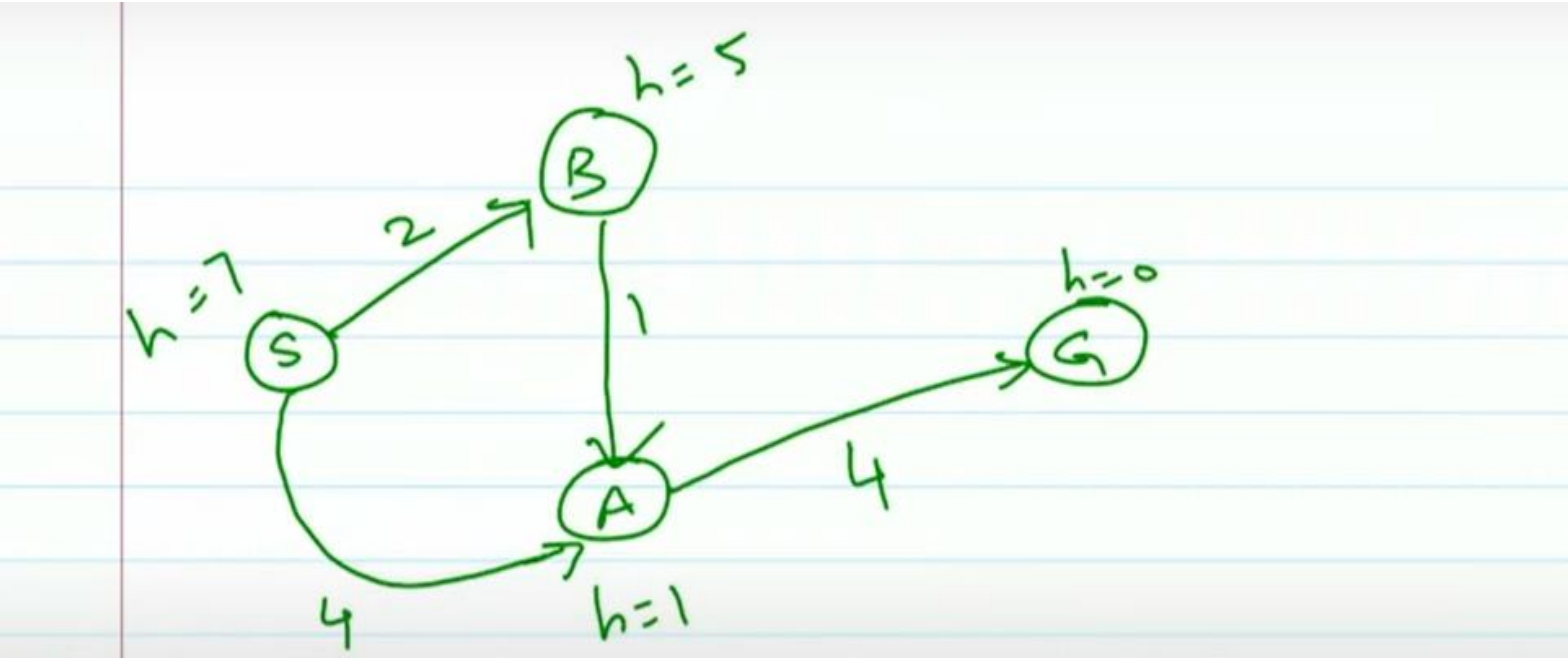
A* Search


Arad
Sibiu
Timisoara
447=118+329
Zerind
449=75+374
Arad
646=280+366
Fagaras
Oradea
671=291+380
Rimnicu Vilcea
Sibiu
591=338+253
Bucharest
450=450+0
Craiova
526=366+160
Pitesti
Sibiu
553=300+253
Bucharest
418=418+0
Craiova
615=455+160
Rimnicu Vilcea
607=414+193

Admissible Heuristics

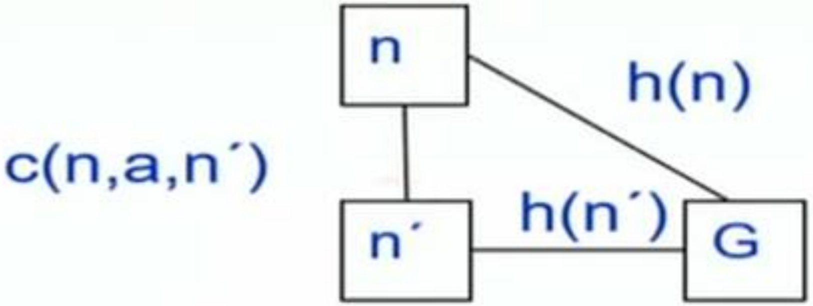
- A heuristic function $h(n)$ is admissible if for every node n, $h(n) \le h^*(n)$, where $h^*(n)$ is the true cost to reach the goal state from n.
- An admissible heuristic never overestimates the cost to reach the goal, i.e., it is optimistic
- Example: $h_{SLD}(n)$ (never overestimates the actual road distance)
- Theorem: If $h(n)$ is admissible, A^* using TREE−SEARCH is optimal

Example



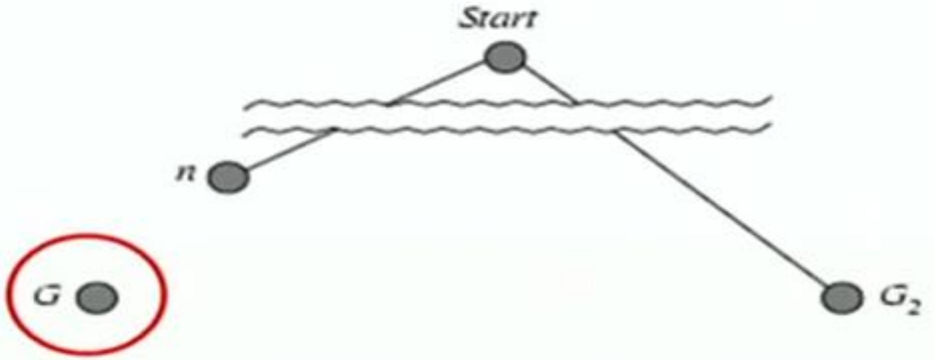
Consistent Heuristic

- h(n) is consistent if
  - for every node n
  - for every successor n´ due to legal action a
  - $h(n) <= c(n,a,n') + h(n')$



Proof of Optimality of A*

- Assume $h()$ is admissible.
  Say some sub-optimal goal state G_2 has been generated and is on the fronti
  Let n be an unexpanded state such that n is on an optimal path to the optim
  goal G.



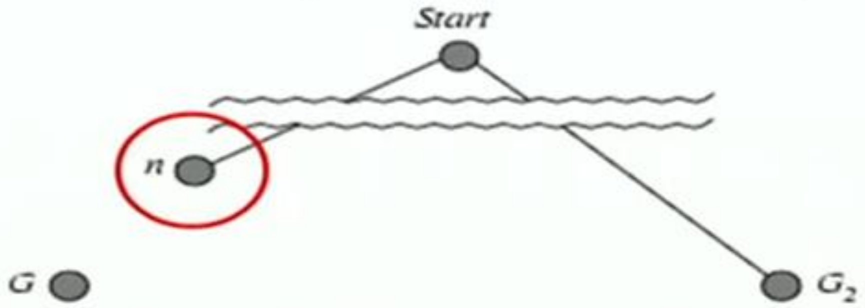
$f(G_2) = g(G_2)$	since $h(G_2) = 0$
$g(G_2) > g(G)$	since G_2 is suboptimal

Focus on G:

$f(G) = g(G)$	since $h(G) = 0$
$f(G_2) > f(G)$	substitution

Proof of Optimality of A*

- Assume *h*() is admissible.
  Say some sub-optimal goal state G_2 has been generated and is on the frontier.
  Let *n* be an unexpanded state such that *n* is on an optimal path to the optimal goal *G*.



$f(G_2) = g(G_2)$ since $h(G_2) = 0$
$g(G_2) > g(G)$ since G_2 is suboptimal

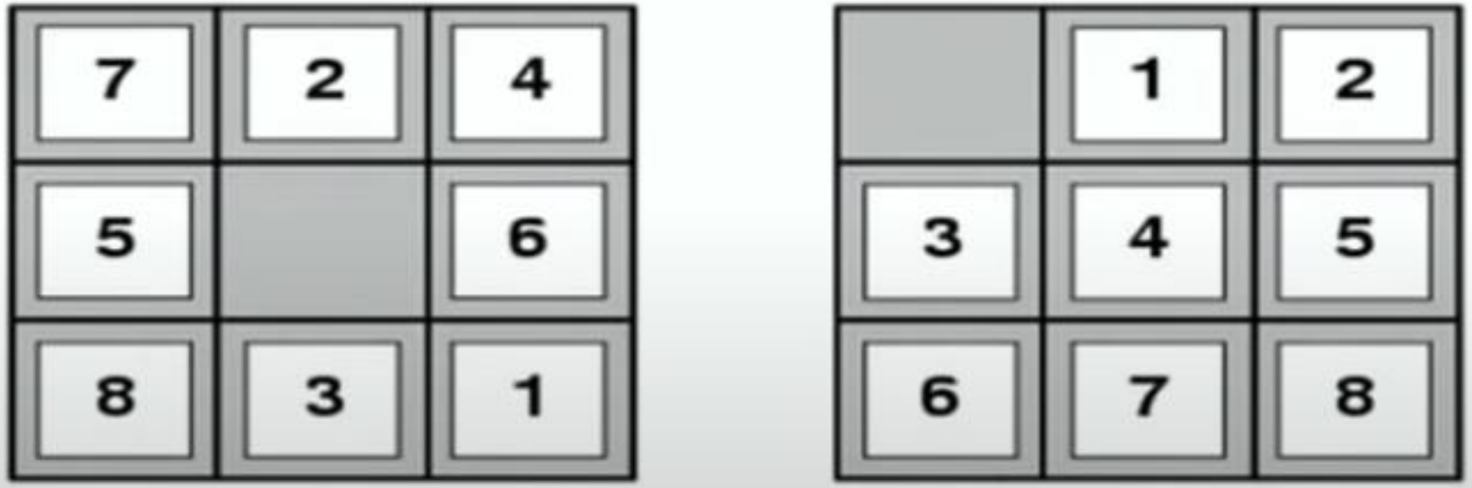
$f(G) = g(G)$ since $h(G) = 0$
$f(G_2) > f(G)$ substitution

Now focus on n:

$h(n) \leq h^*(n)$ since h is admissible
$g(n) + h(n) \leq g(n) + h^*(n)$ algebra
$f(n) = g(n) + h(n)$ definition
$f(G) = g(n) + h^*(n)$ by assum
$f(n) \leq f(G)$ substitution

Hence $f(G_2) > f(n)$, and A^* will never select G_2 for expansion.

Admissible Heuristics



Start State

Goal State

Admissible Heuristics

E.g., for the 8-puzzle:

- $h_1(n)$ = number of misplaced tiles
- $h_2(n)$ = total Manhattan distance

(i.e., no. of squares from desired location of each tile)

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

- $h_1(S) = ?$ 8
- $h_2(S) = ?$ 3+1+2+2+2+3+3+2 = 18

Dominance

- If $h_2(n) \geq h_1(n)$ for all n (both admissible) then h_2 dominates h_1
- h_2 is better for search

- Typical search costs (average number of node expanded):

- $d=12$ IDS = 3,644,035 nodes
  $A^*(h_1)$ = 227 nodes
  $A^*(h_2)$ = 73 nodes
- $d=24$ IDS = too many nodes
  $A^*(h_1)$ = 39,135 nodes
  $A^*(h_2)$ = 1,641 nodes