

Artificial Intelligence

Problem solving by searching: Informed Search

by

Abdul Sahli Fakhardin

Faculty Computer Systems & Software Engineering

sahli@ump.edu.my

Chapter Description

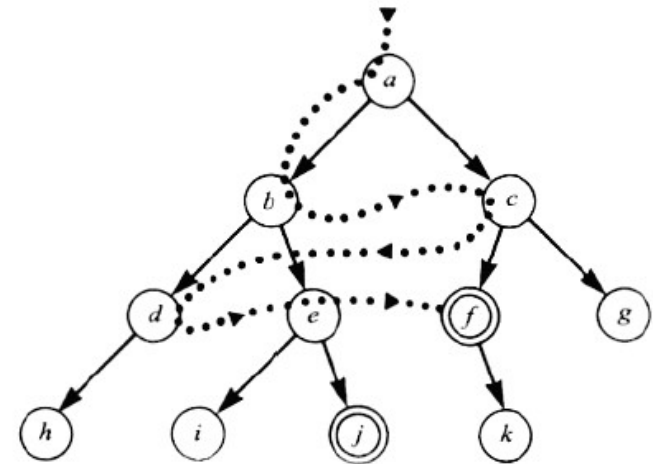
- Expected Outcomes
 - Student able to apply the greedy search to solve given problem
 - Student able to apply the A* search to solve given problem
- References
 -

Content #1

- Heuristic function
- Greedy search
- A* search

Heuristic function

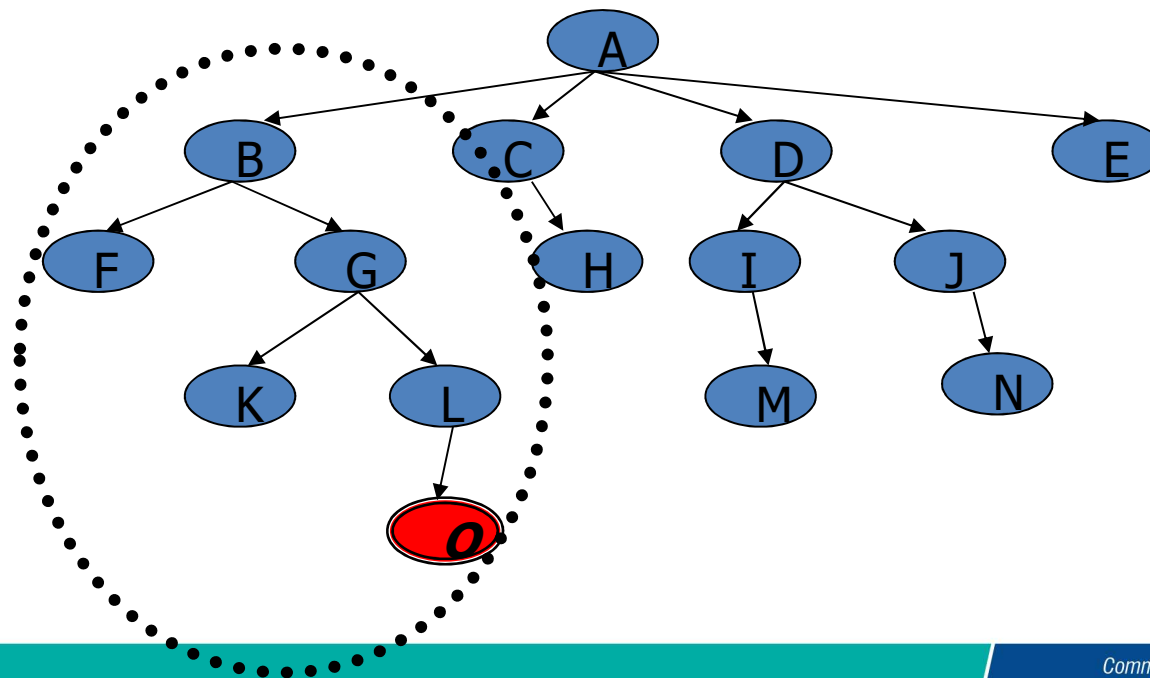
- Without incorporating some knowledge into the search, it does not have any preference on the search space.
- The search look everywhere to find the answer.



Heuristic function

- With the knowledge, the search space is given “hints” when searching the problem.
 - Heuristic information in search = Hints

Search in this subtree



Heuristic function

- A heuristic function $f(n)$, gives an estimation on the “cost” of getting from node n to the goal state – so that the node with the least cost among all possible choices can be selected for expansion first.

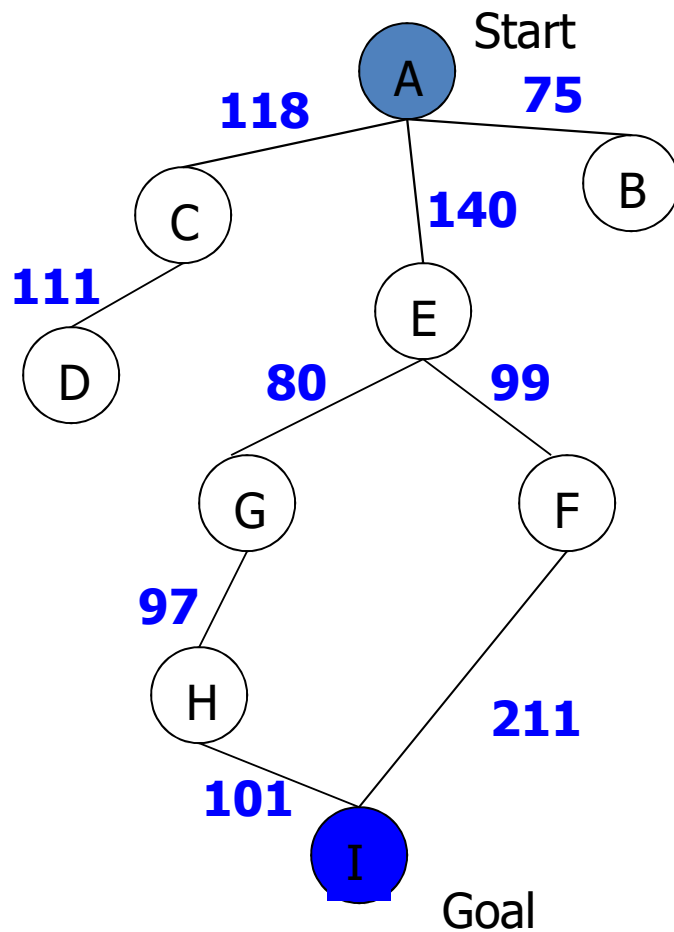
Informed search strategies

- Greedy search
 - Heuristic function $f(n) = h(n)$
- A* search
 - Heuristic function $f(n) = h(n) + g(n)$

Informed Search Strategies

Greedy Search
eval-fn: $f(n) = h(n)$

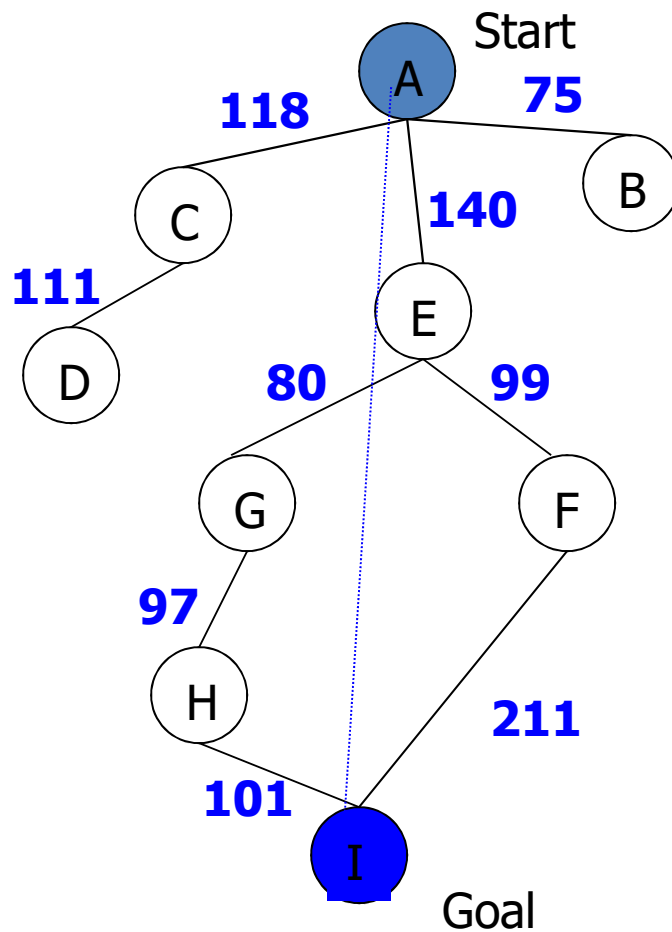
Greedy Search



State	Heuristic: $h(n)$
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

$f(n) = h(n) =$ straight-line distance heuristic

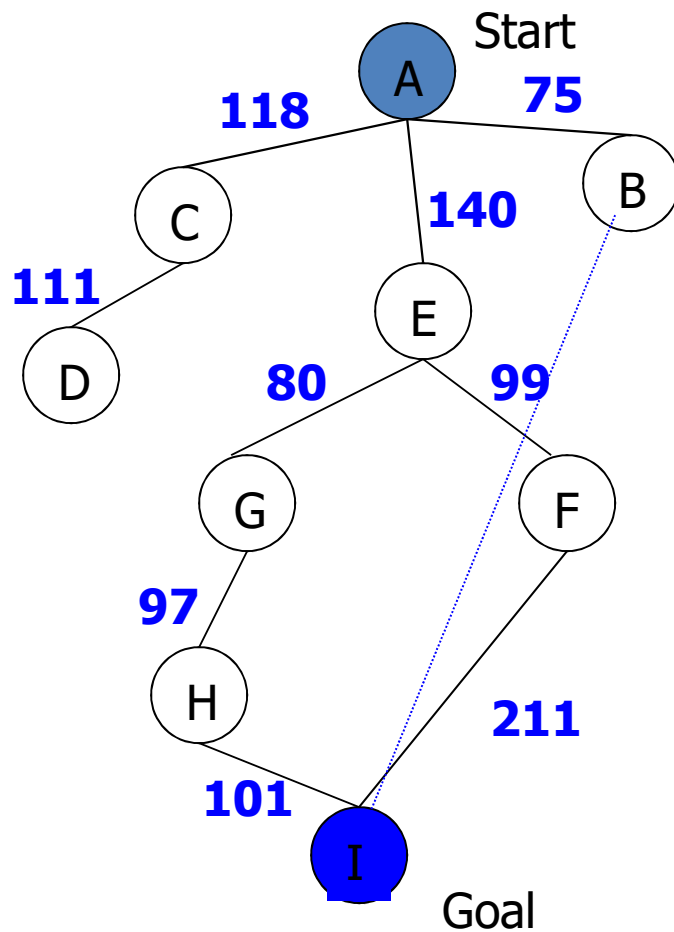
Greedy Search



State	Heuristic: $h(n)$
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

$f(n) = h(n) =$ straight-line distance heuristic

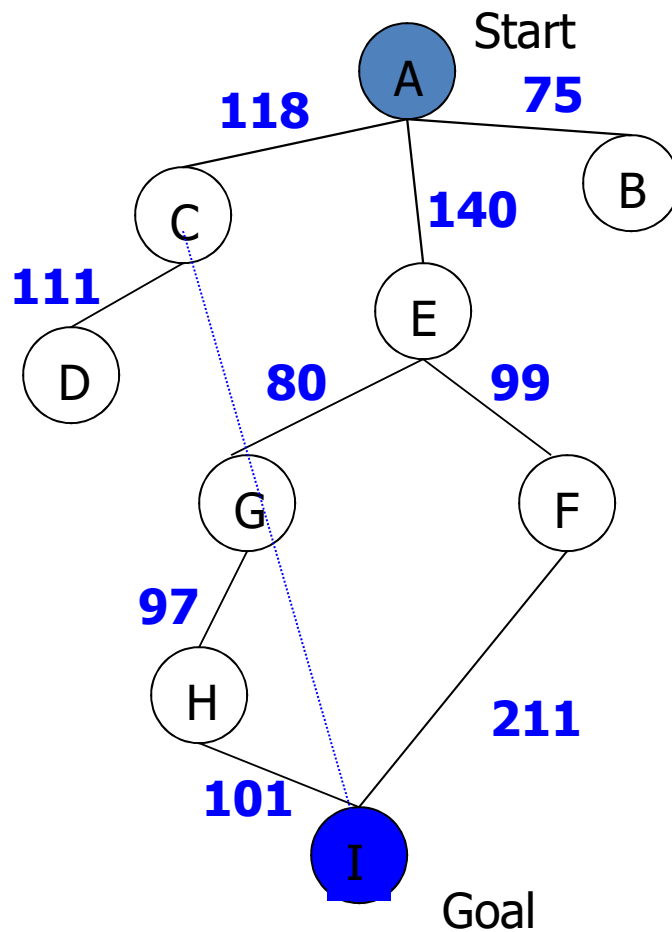
Greedy Search



State	Heuristic: $h(n)$
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

$f(n) = h(n) =$ straight-line distance heuristic

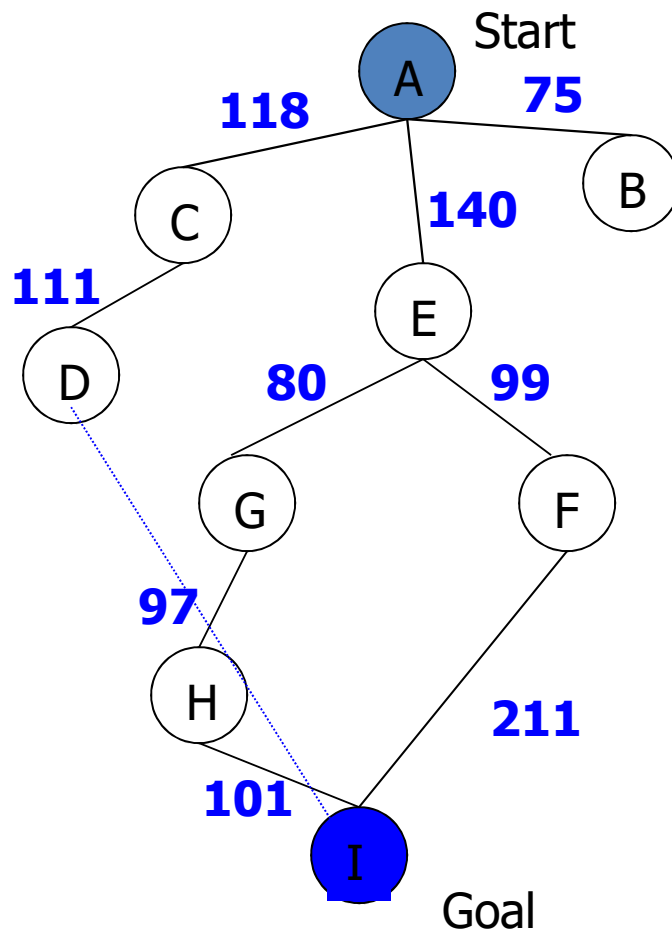
Greedy Search



State	Heuristic: $h(n)$
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

$f(n) = h(n) =$ straight-line distance heuristic

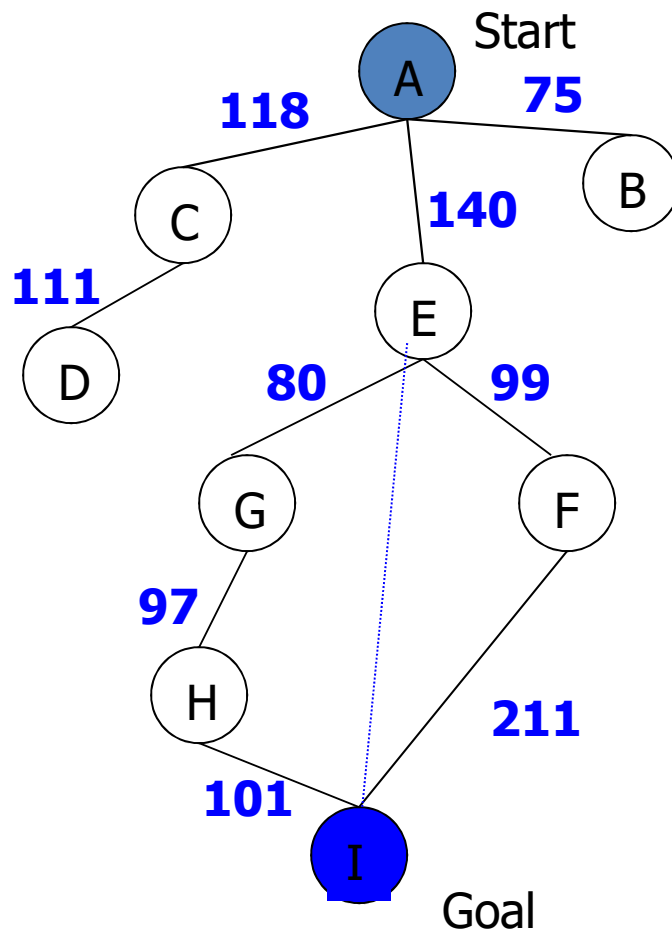
Greedy Search



State	Heuristic: $h(n)$
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

$f(n) = h(n) =$ straight-line distance heuristic

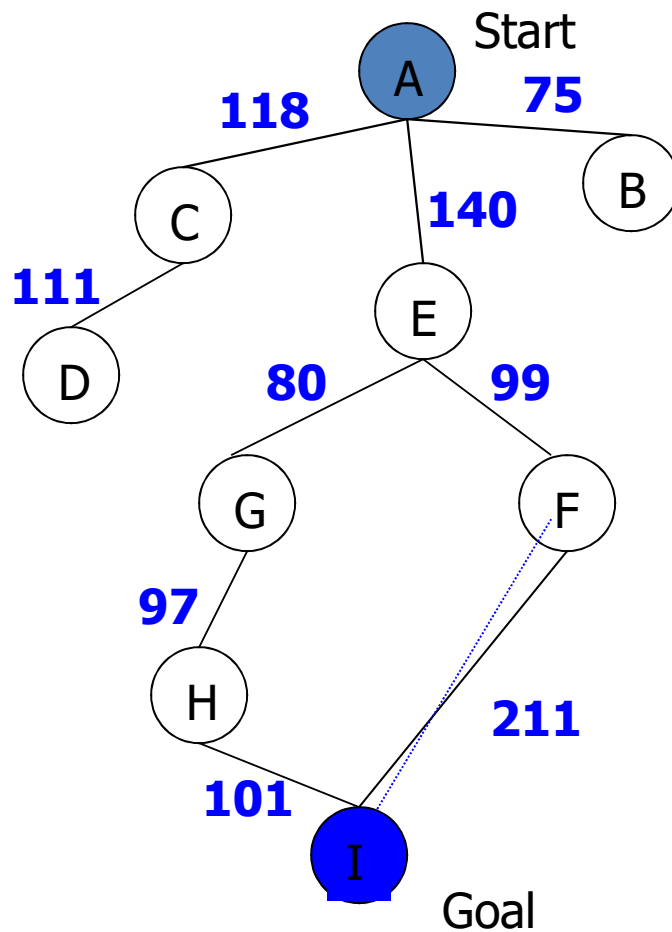
Greedy Search



State	Heuristic: $h(n)$
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

$f(n) = h(n) =$ straight-line distance heuristic

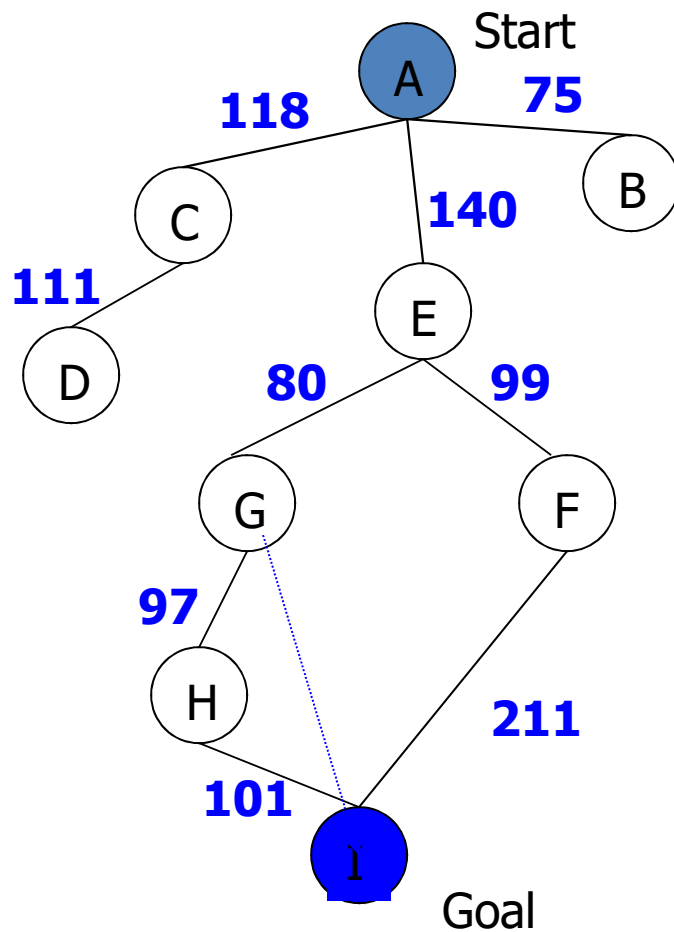
Greedy Search



State	Heuristic: $h(n)$
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

$f(n) = h(n) =$ straight-line distance heuristic

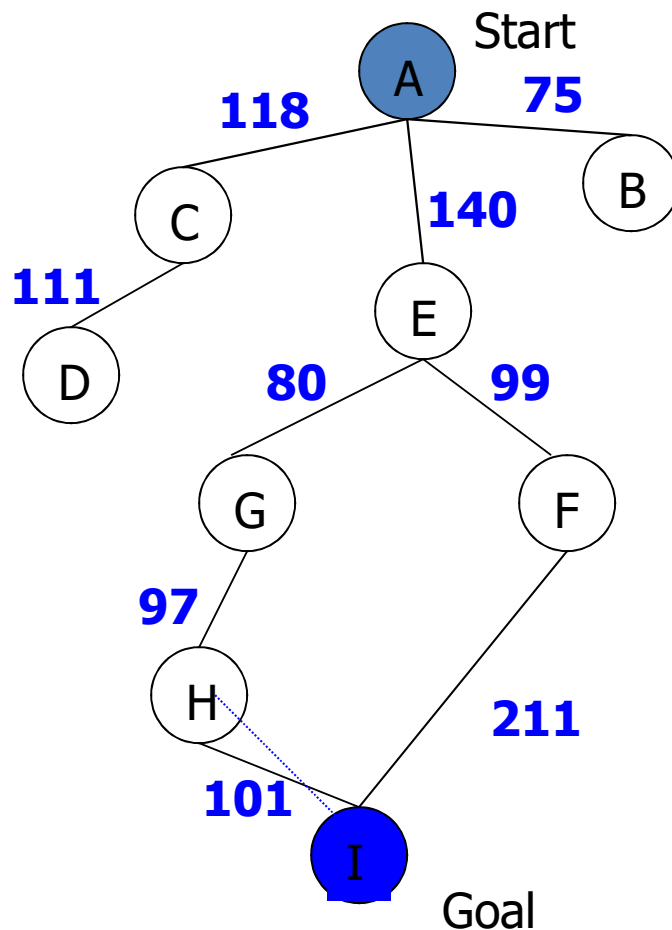
Greedy Search



State	Heuristic: $h(n)$
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

$f(n) = h(n) =$ straight-line distance heuristic

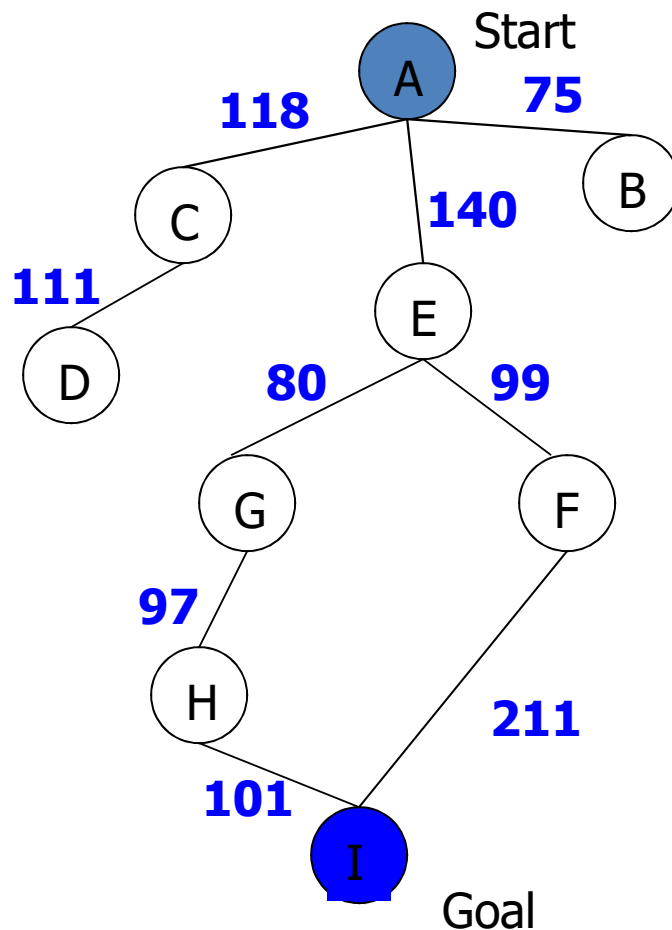
Greedy Search



State	Heuristic: $h(n)$
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

$f(n) = h(n) =$ straight-line distance heuristic

Greedy Search



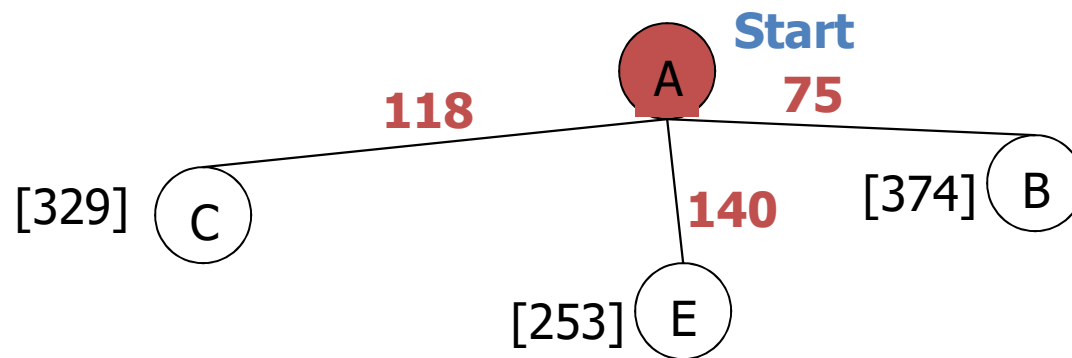
State	Heuristic: $h(n)$
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

$f(n) = h(n) =$ straight-line distance heuristic

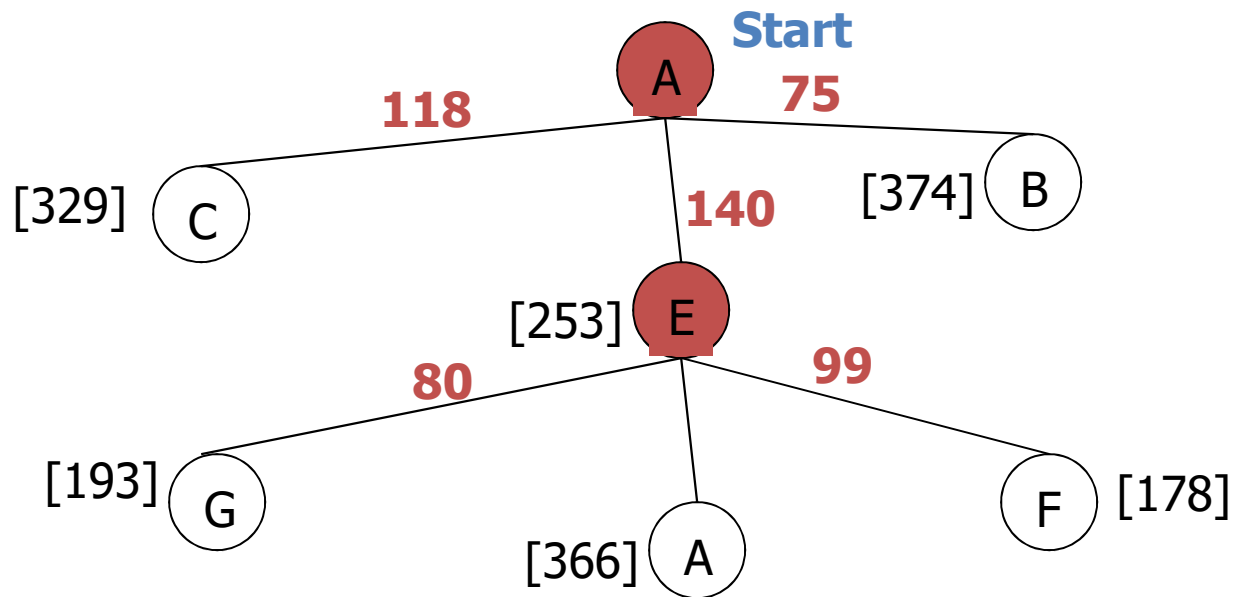
Greedy Search: Tree Search

A **Start**

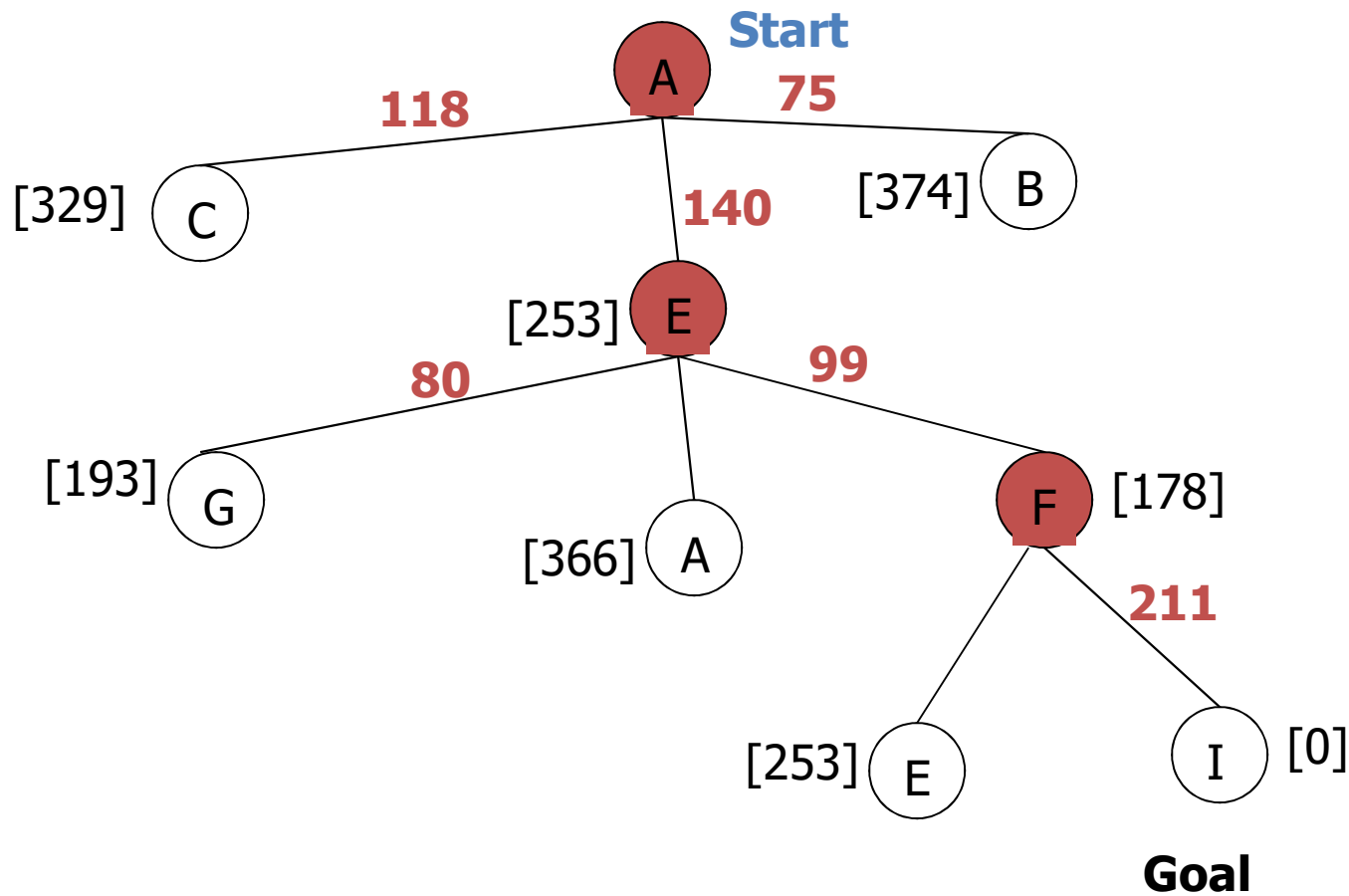
Greedy Search: Tree Search



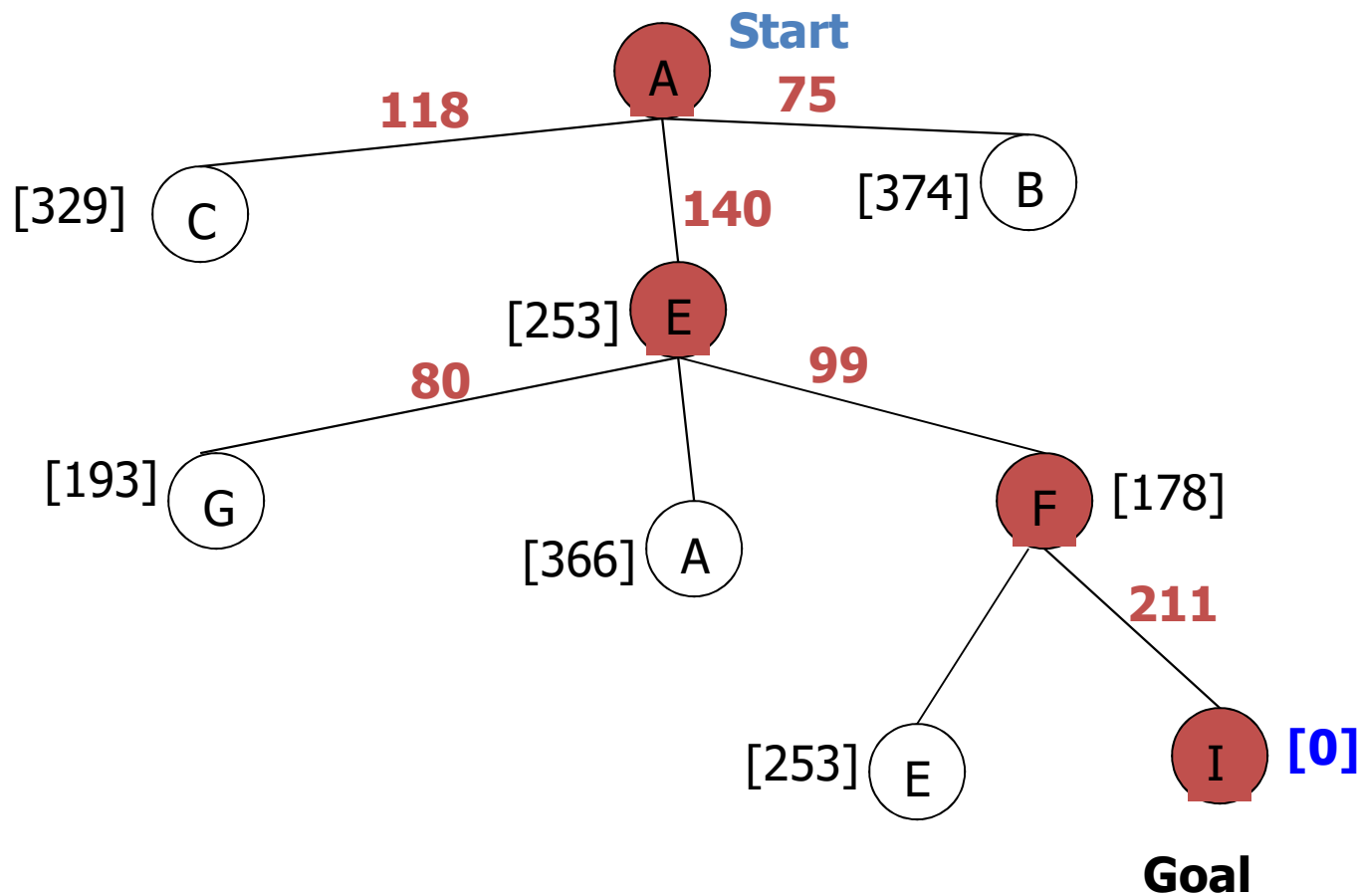
Greedy Search: Tree Search



Greedy Search: Tree Search



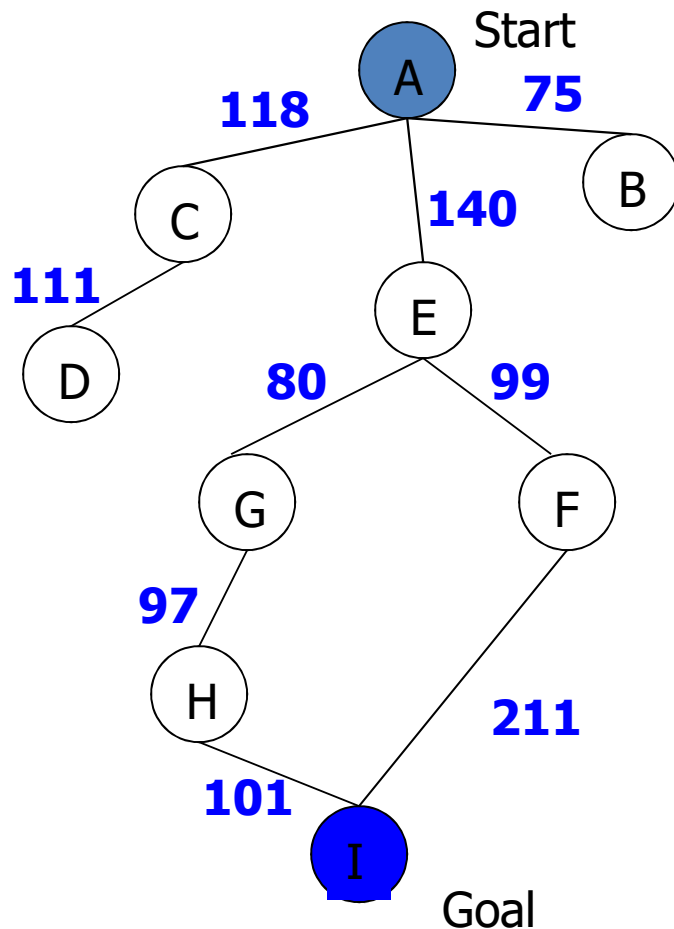
Greedy Search: Tree Search



Path cost(A-E-F-I) = 253 + 178 + 0 = 431

dist(A-E-F-I) = 140 + 99 + 211 = 450

Greedy Search: Time and Space Complexity ?

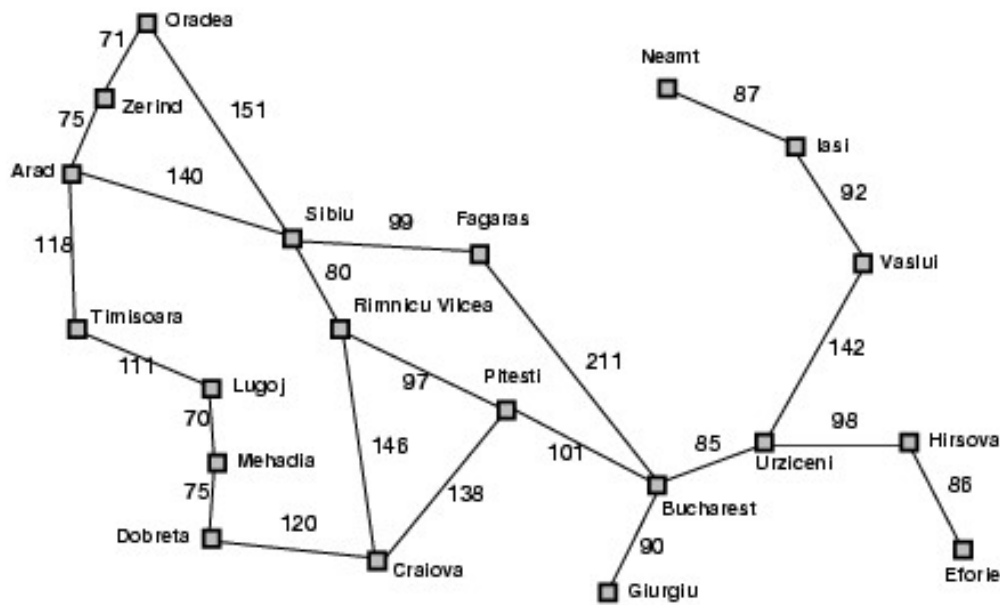
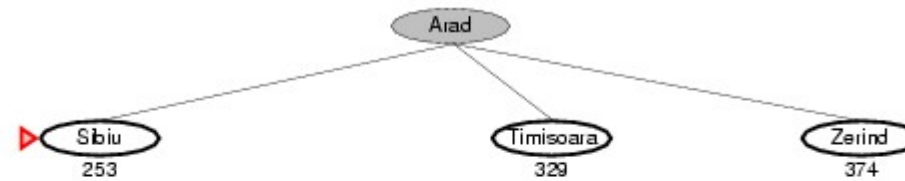


- Greedy search is not optimal.
- Greedy search is incomplete **without systematic checking of repeated states.**

- In the worst case, the Time and Space Complexity of Greedy Search are both $O(b^m)$

Where b is the branching factor and m the maximum path length

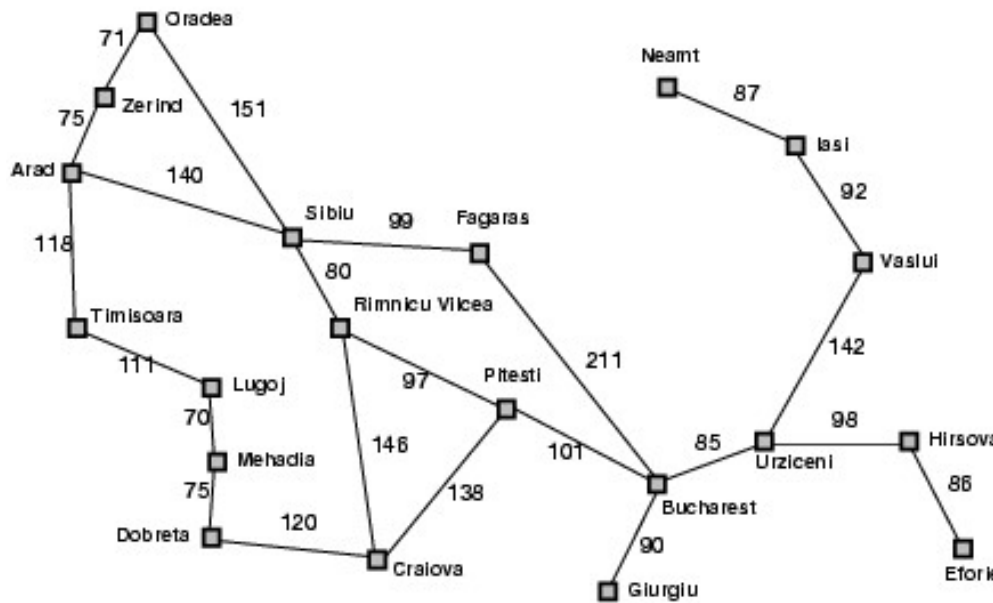
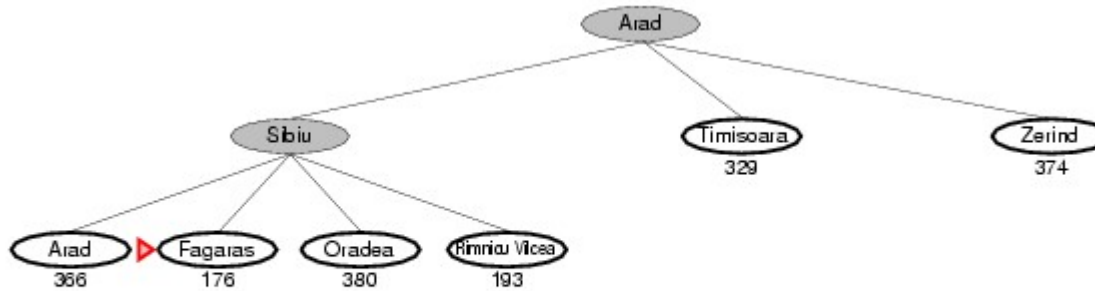
Greedy best-first search example



Straight-line distance
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

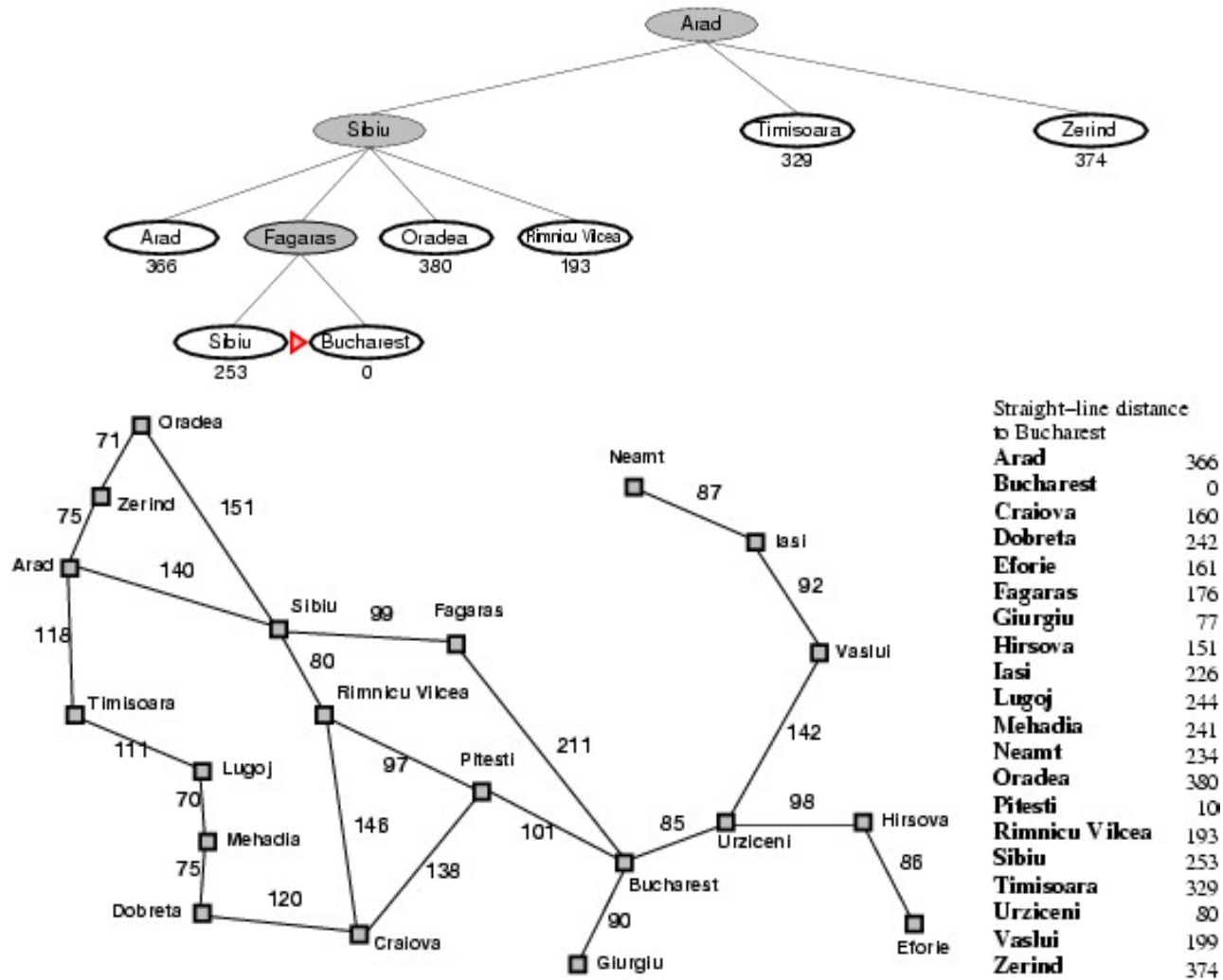
Greedy best-first search example



Straight-line distance
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Greedy best-first search example



Informed Search Strategies

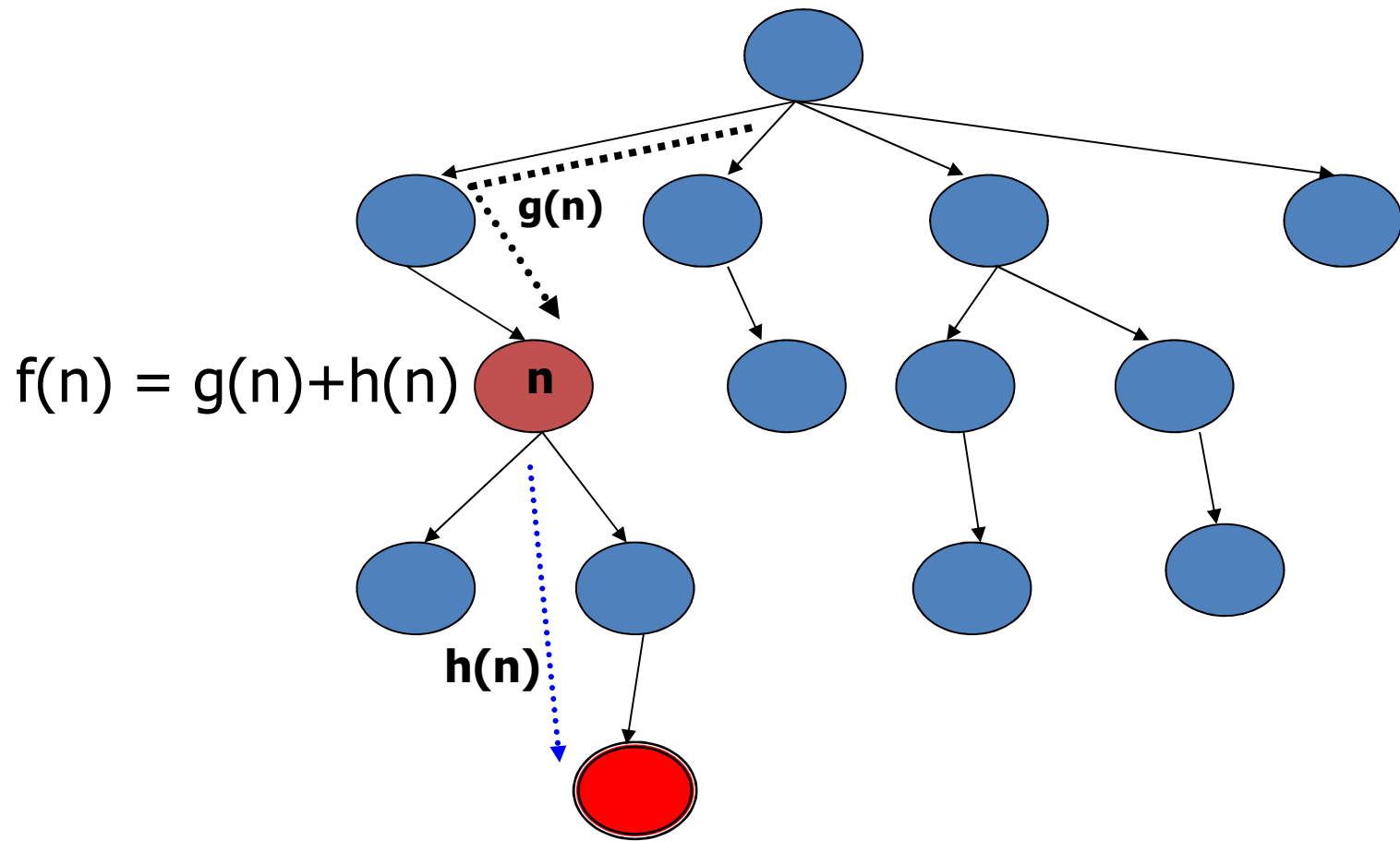
A* Search

eval-fn: $f(n) = g(n) + h(n)$

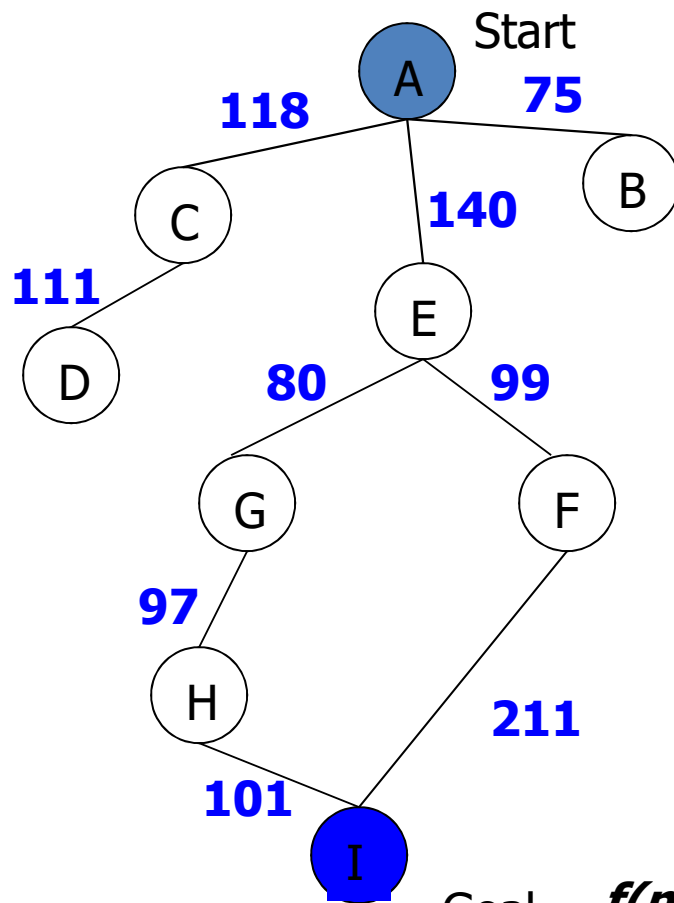
A* (A Star)

- A* uses a heuristic function which combines $g(n)$ and $h(n)$: $f(n) = g(n) + h(n)$
- **$g(n)$** is the exact cost to reach node n from the initial state.
- **$h(n)$** is an estimation of the remaining cost to reach the goal.

A* (A Star)



A* Search



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

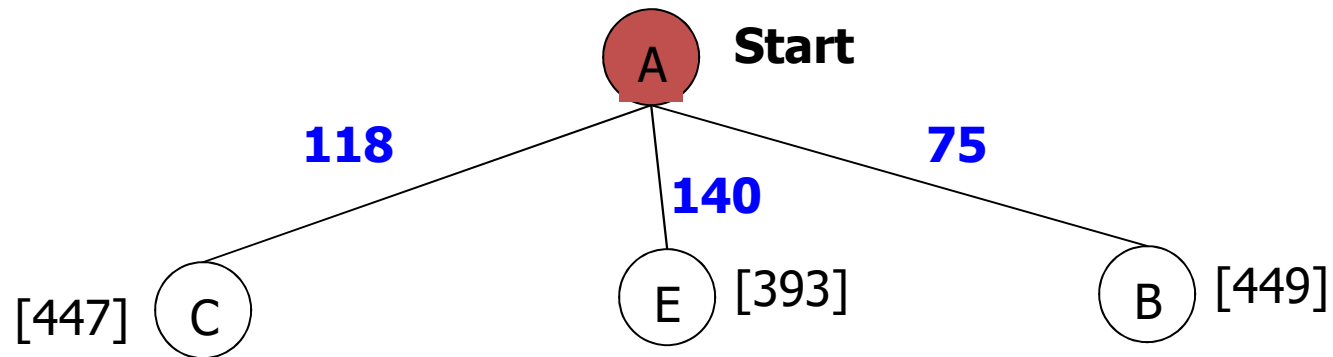
State	Heuristic: $h(n)$
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

$g(n)$: is the exact cost to reach node n from the initial state?

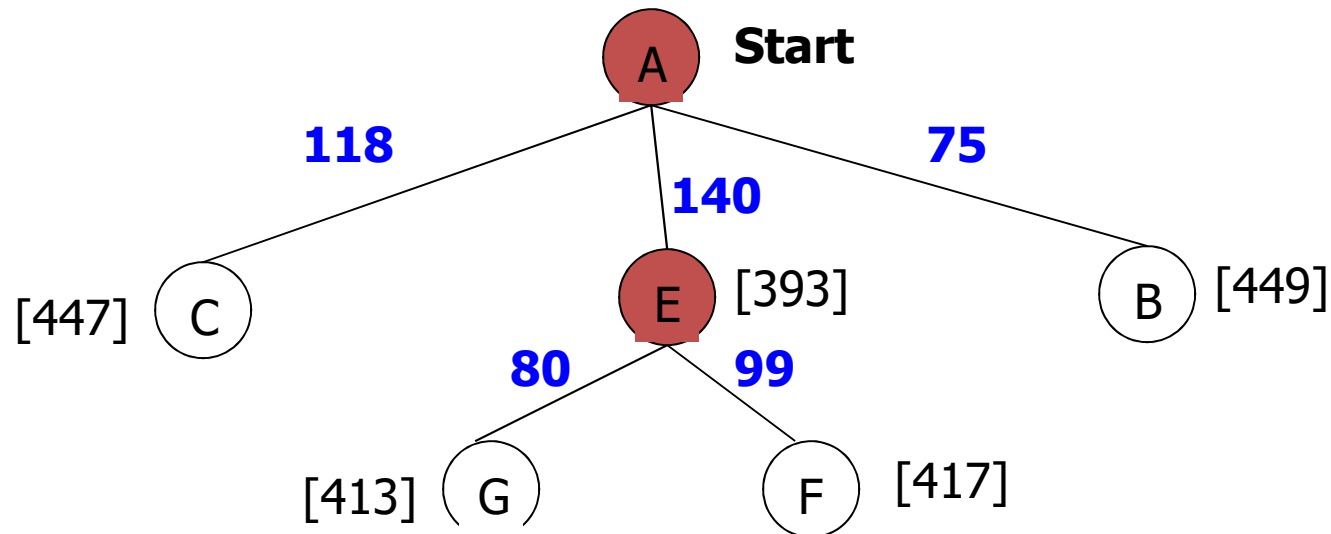
A* Search: Tree Search

A **Start**

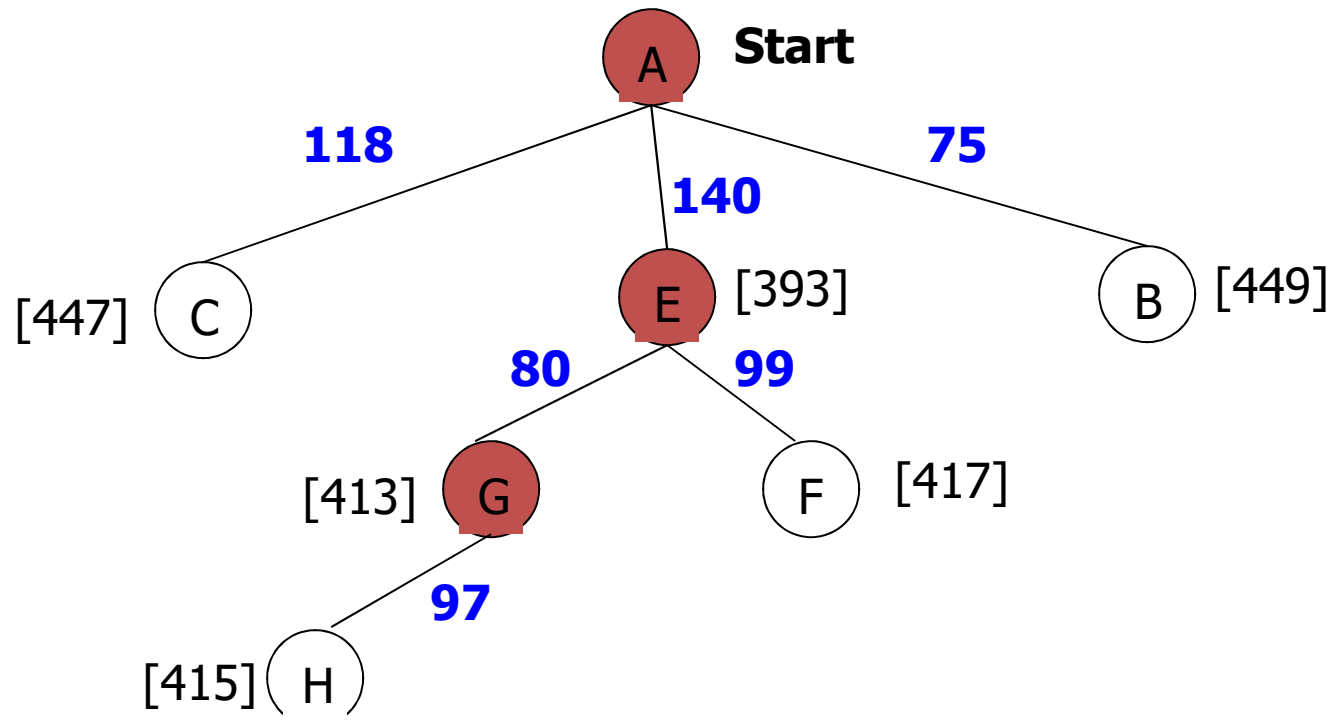
A* Search: Tree Search



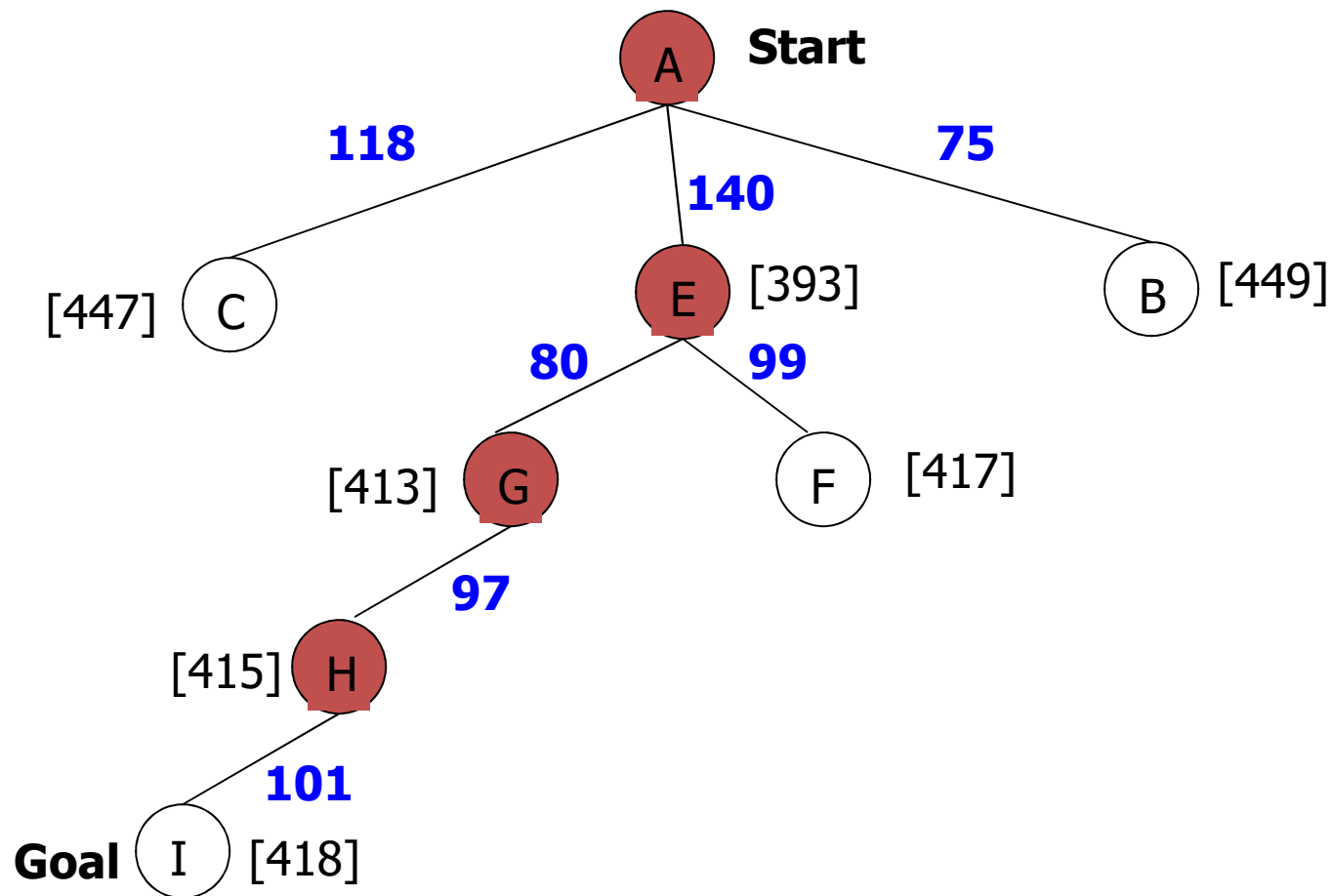
A* Search: Tree Search



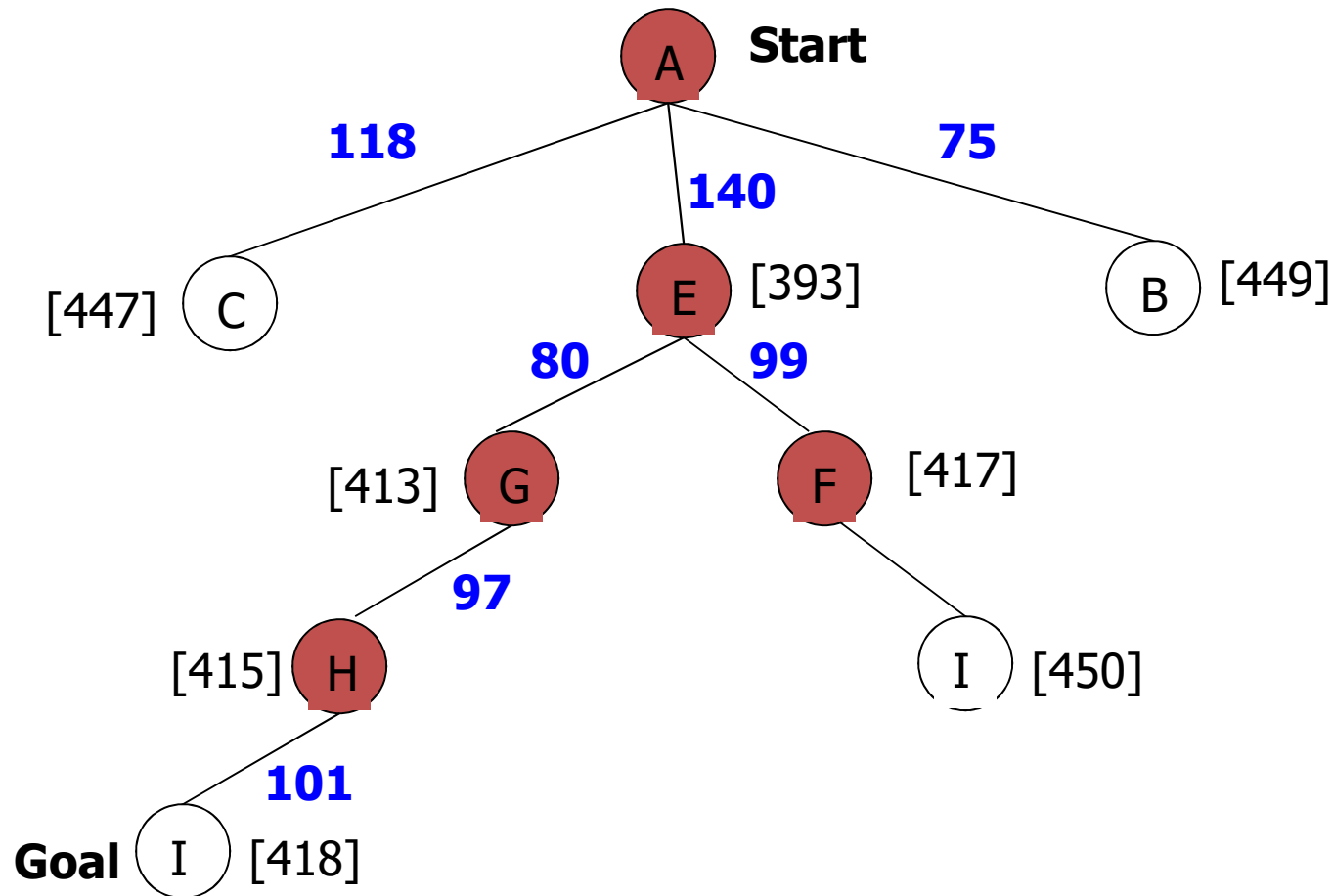
A* Search: Tree Search



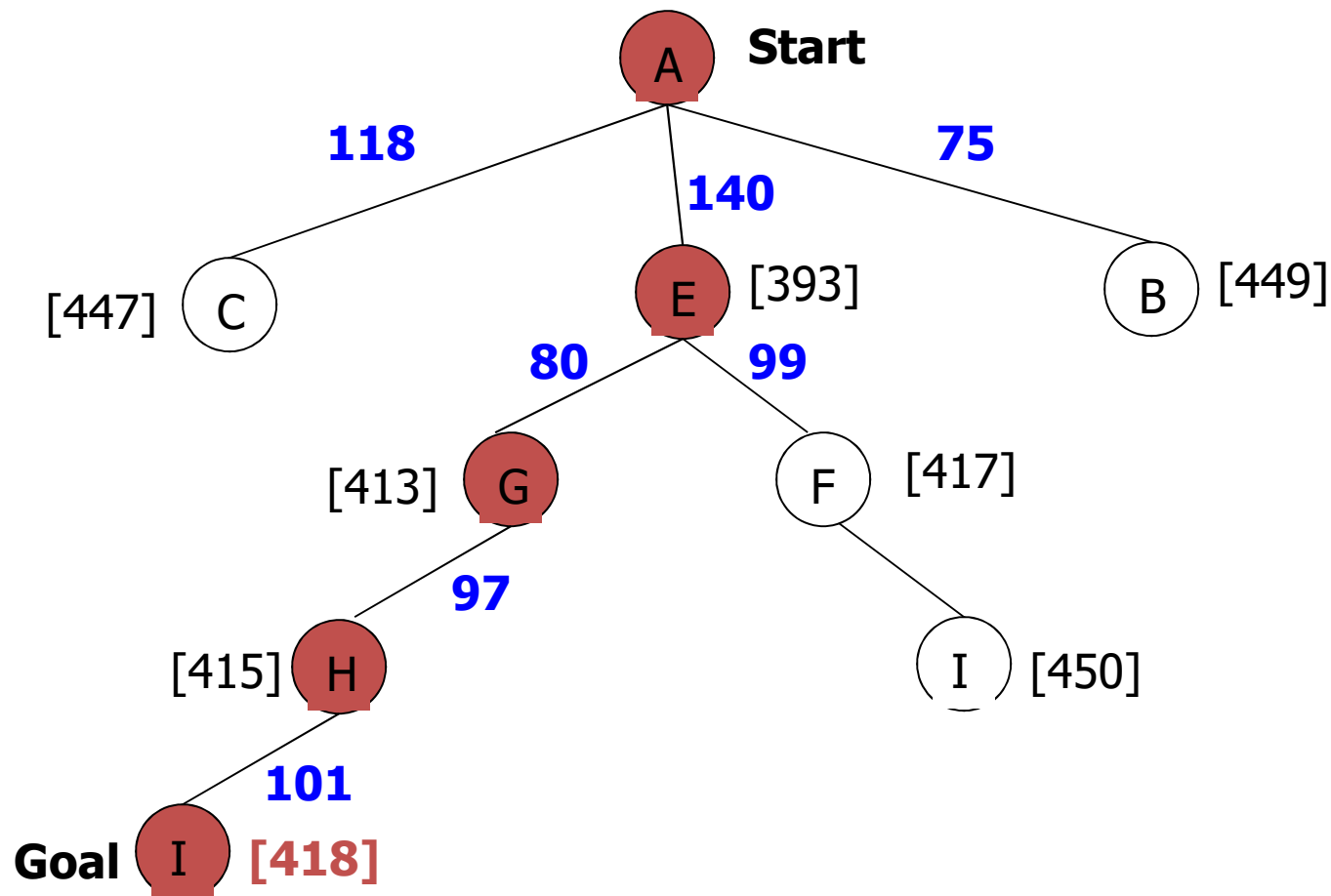
A* Search: Tree Search



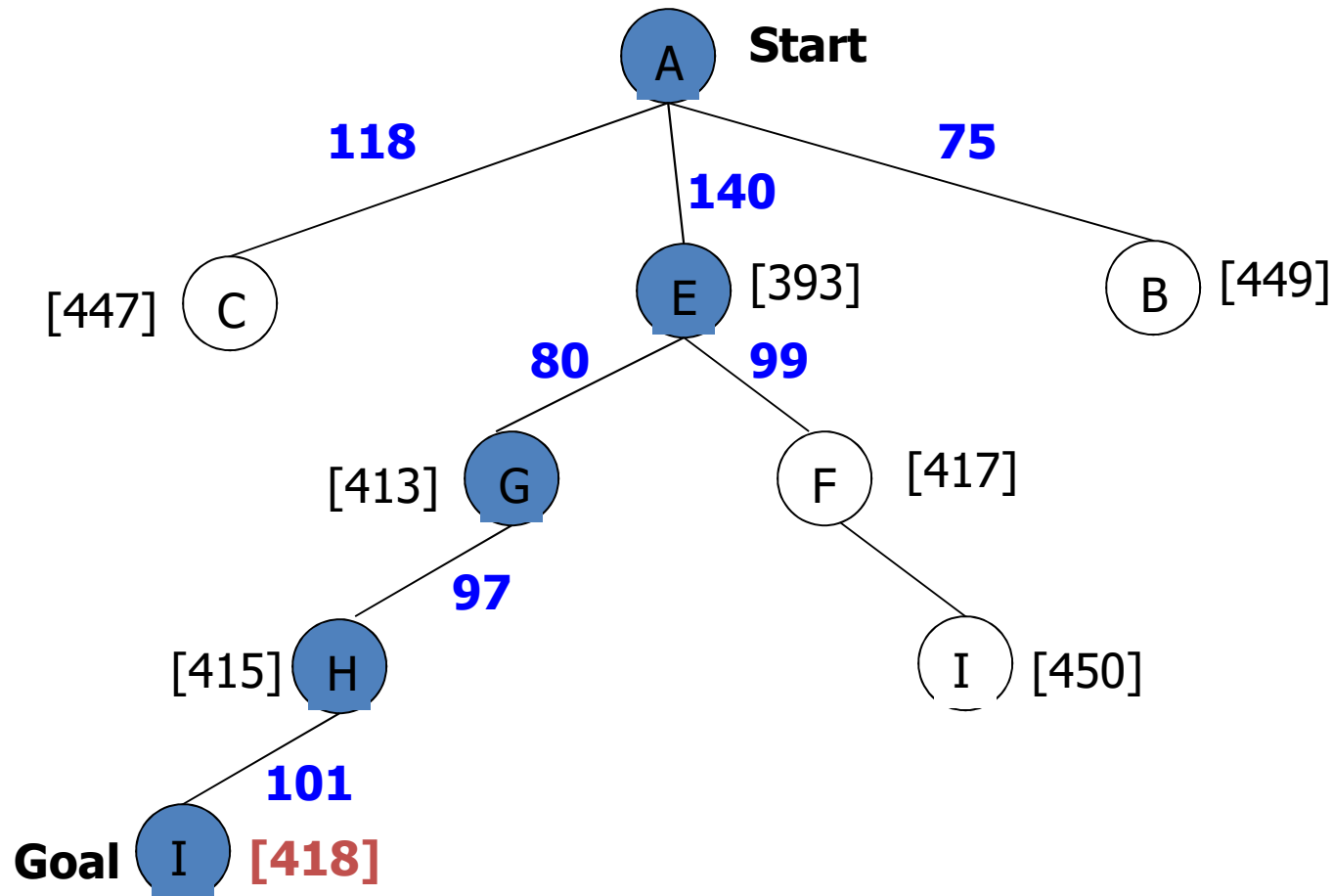
A* Search: Tree Search



A* Search: Tree Search

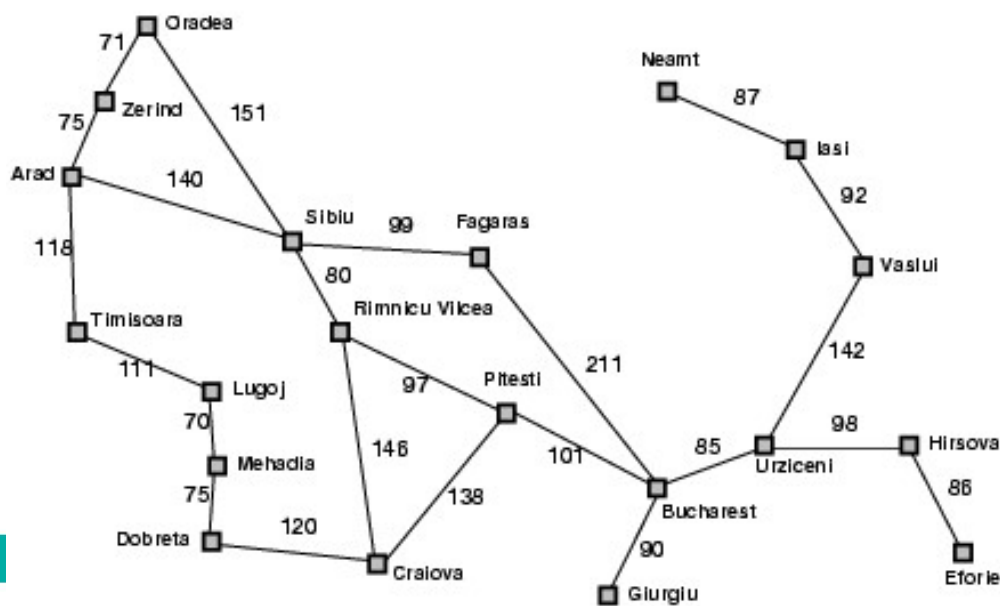


A* Search: Tree Search



A* search example

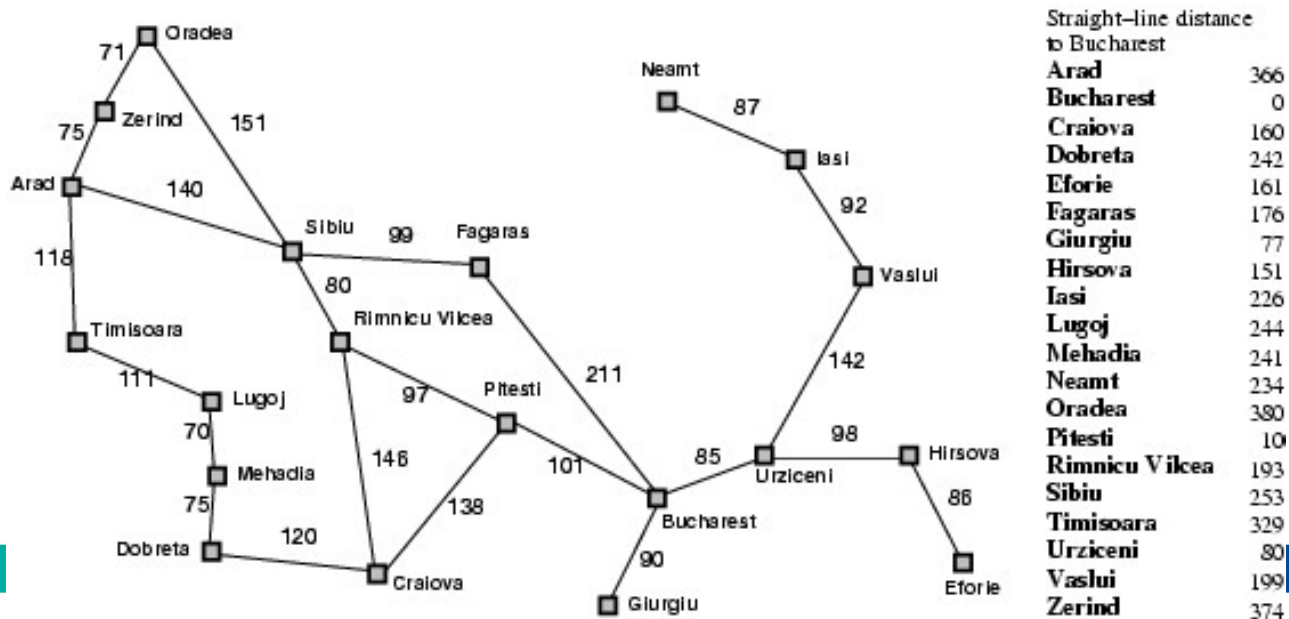
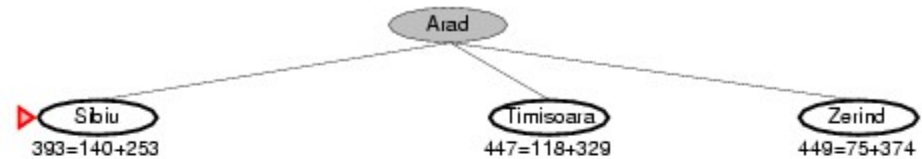
Arad
366=0+366



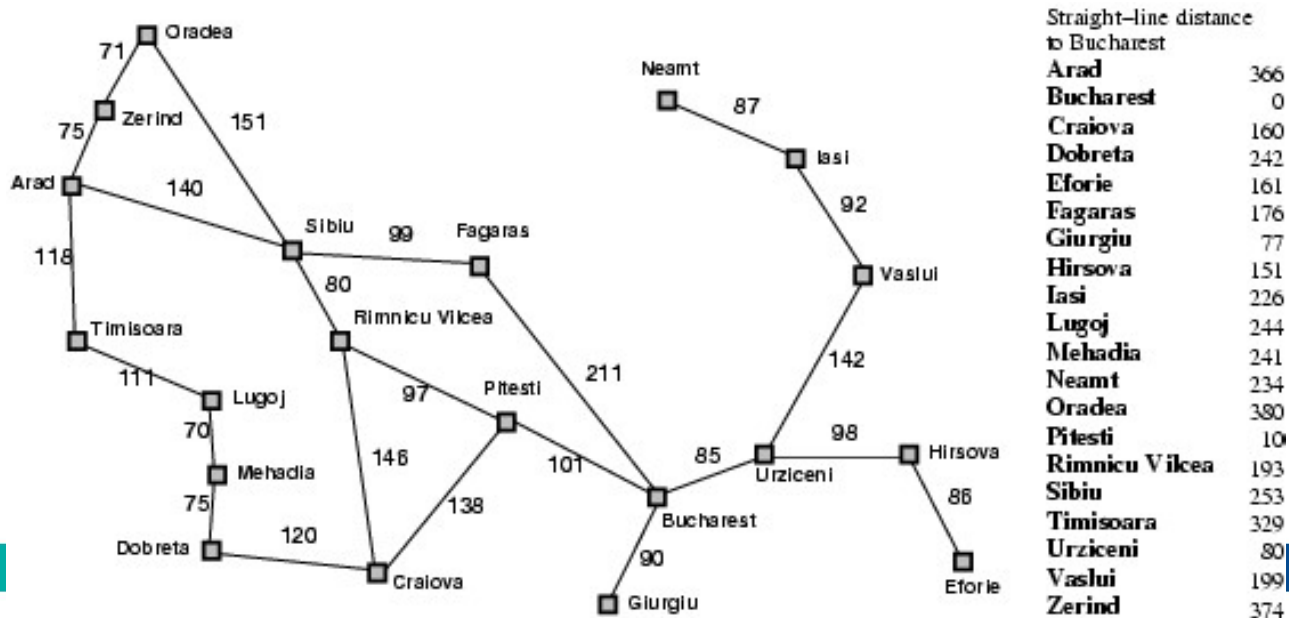
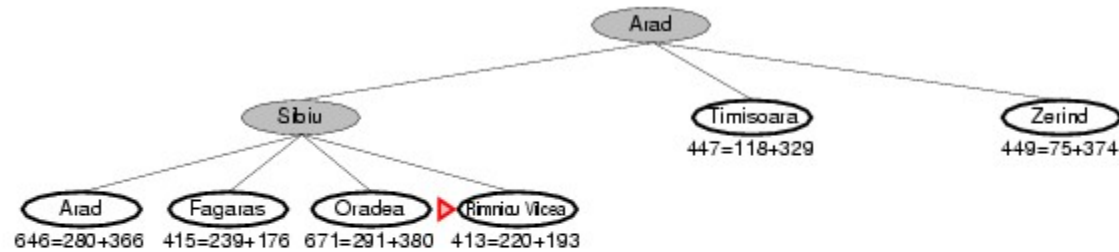
Straight-line distance
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

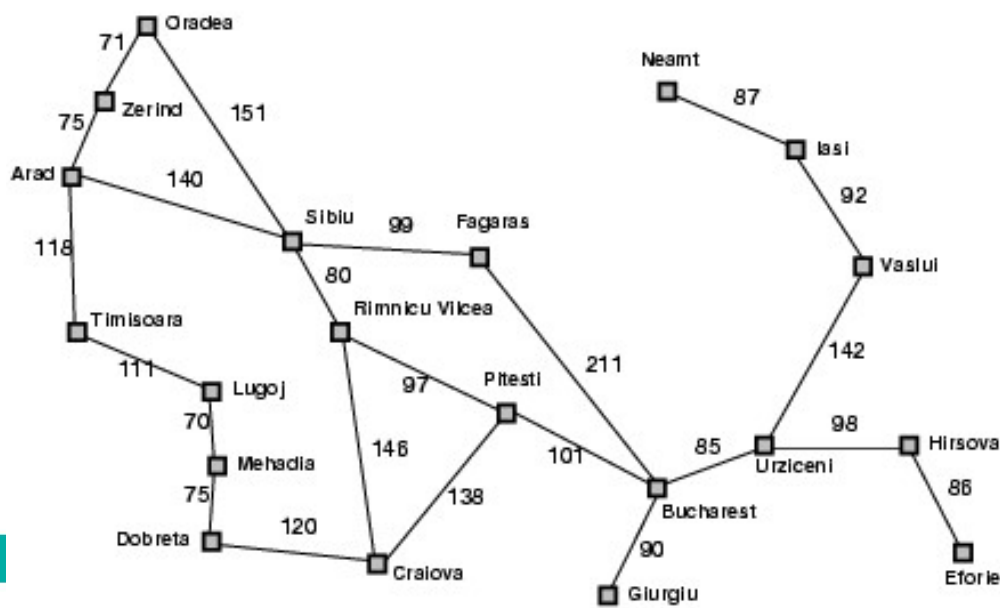
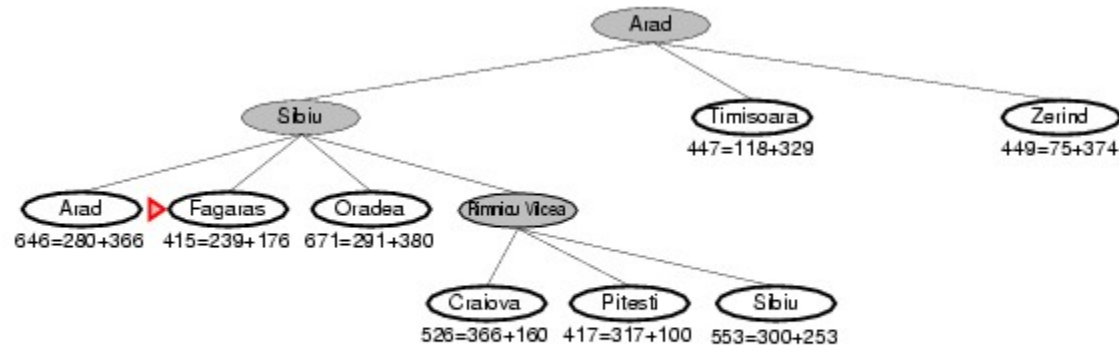
A* search example



A* search example



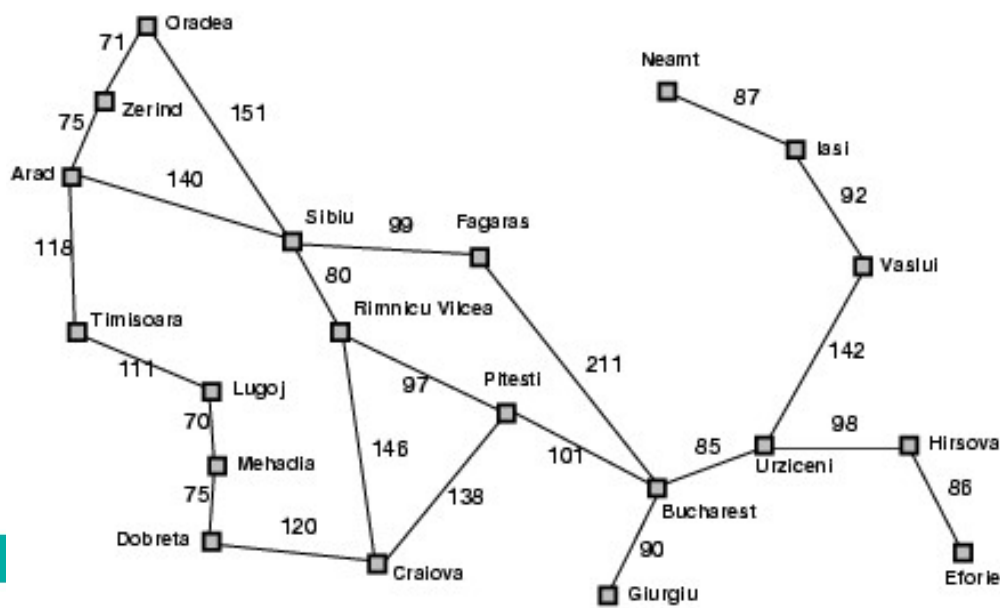
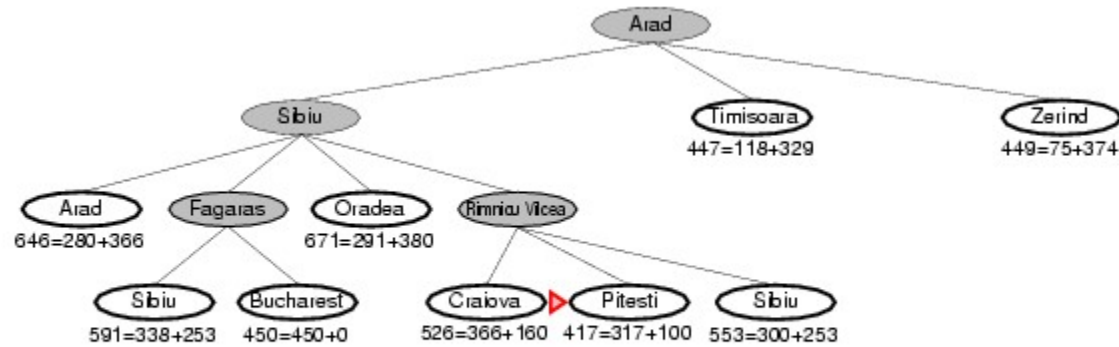
A* search example



Straight-line distance to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

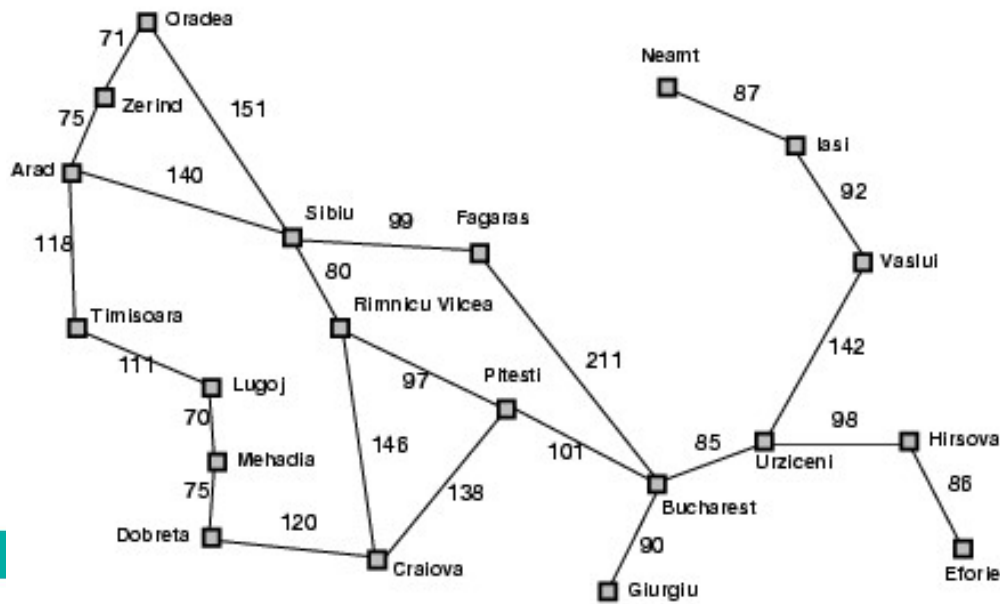
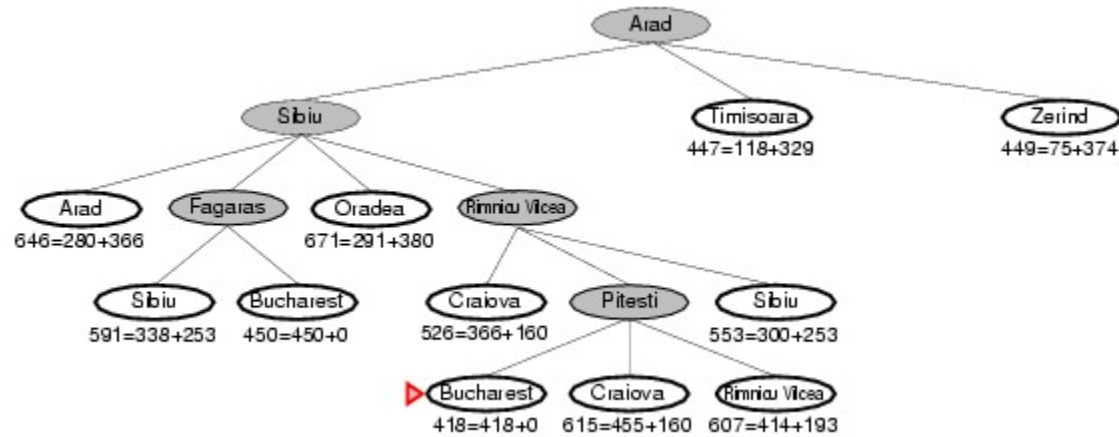
A* search example



Straight-line distance
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

A* search example



Straight-line distance
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Conclusion of The Chapter

- Conclusion #1
 - Informed search used heuristic information to help the search function
- Conclusion #2
 - Greedy search used only the heuristic information to determine the route
- Conclusion #3
 - A* search used heuristic and cost information to determine the route