

CS 480

Introduction to Artificial Intelligence

September 13, 2022

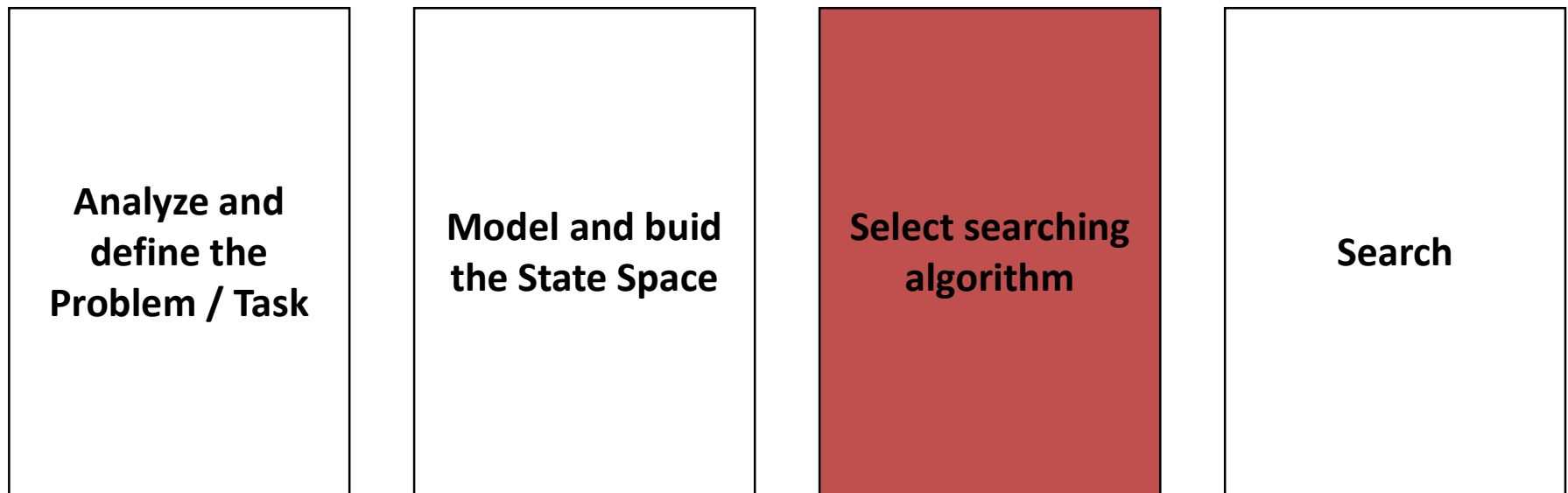
Announcements / Reminders

- Please follow the Week 03 To Do List instructions
- Quiz #01 due on Sunday (09/18/22) at 11:00 PM CST
- Written Assignment #01 due on Tuesday (09/20/22) at 11:00 PM CST
- **Midterm** Exam (consider fixed):
 - October 13th, 2022 during (Thursday) lecture time

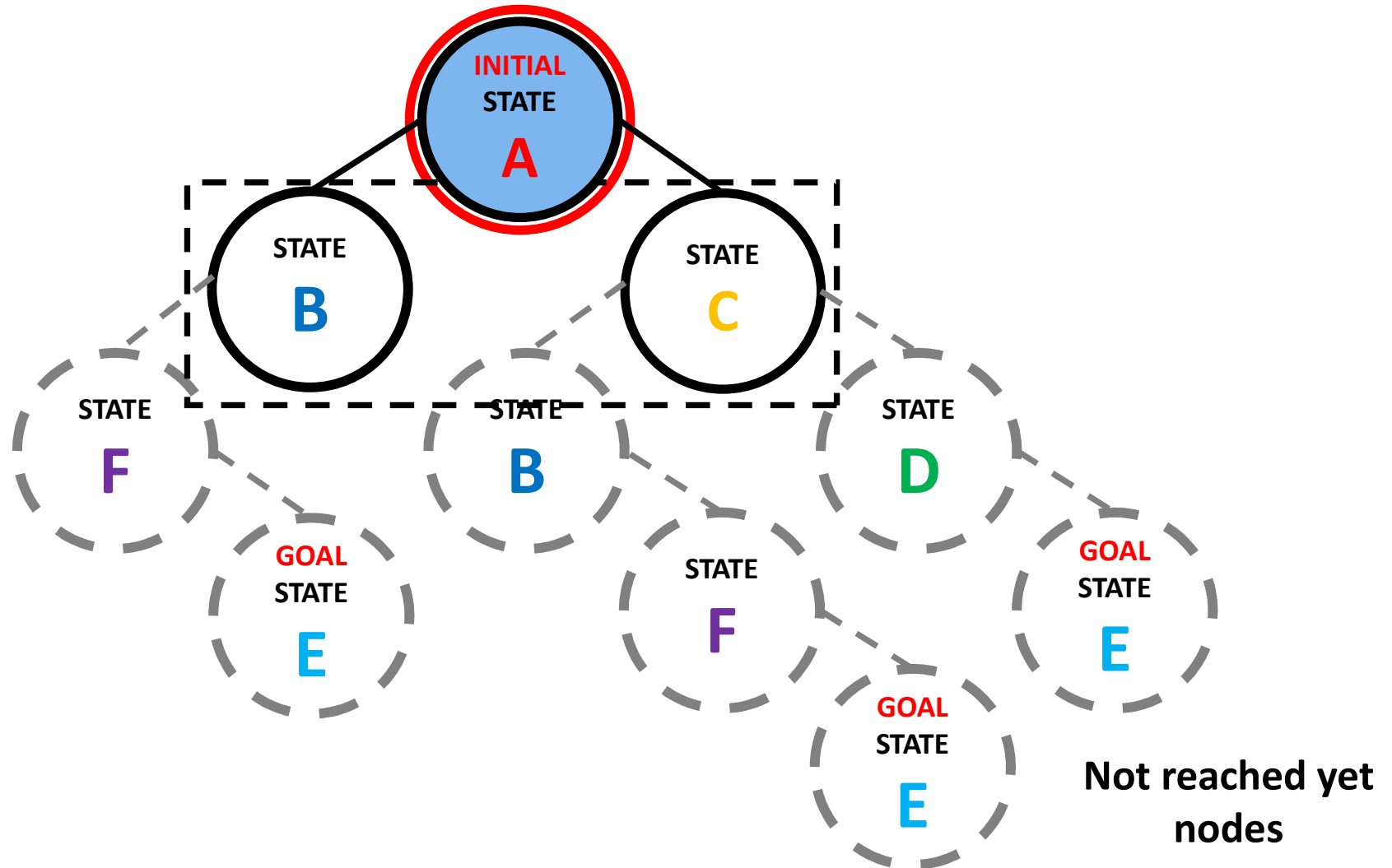
Plan for Today

- **Problem Solving: Searching**
 - **Informed search**
 - **Hill Climbing algorithm**
 - **Greedy Best First algorithm**
 - **A* algorithm**

Designing the Searching Problem



Expansion: Which Node to Expand?



Evaluation function

Calculate / obtain:

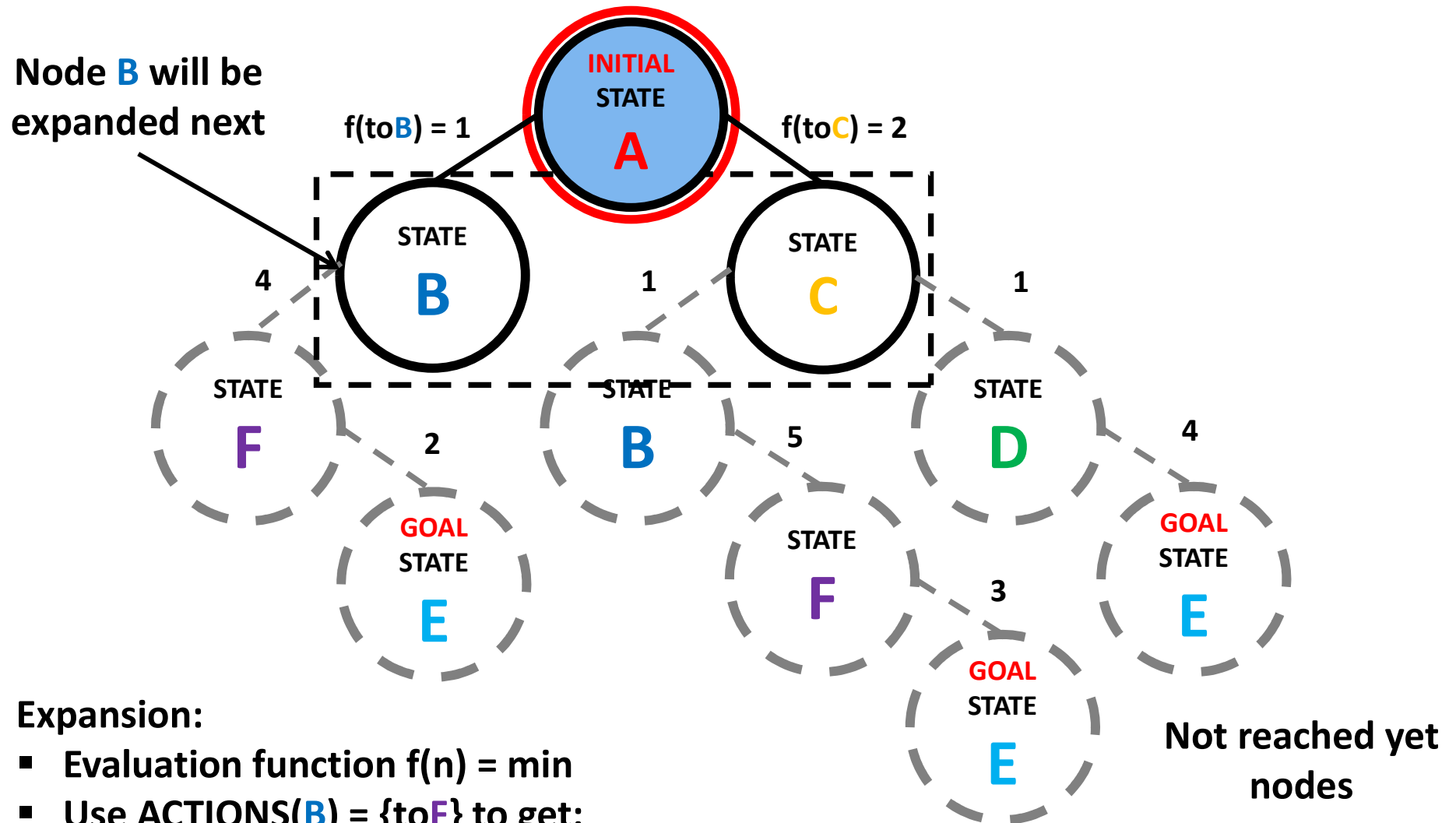
$$f(n) = f(\text{State } n)$$

$$f(n) = f(\text{relevant information about State } n)$$

**A state n with minimum $f(n)$ should be
chosen for expansion**

What about ties?

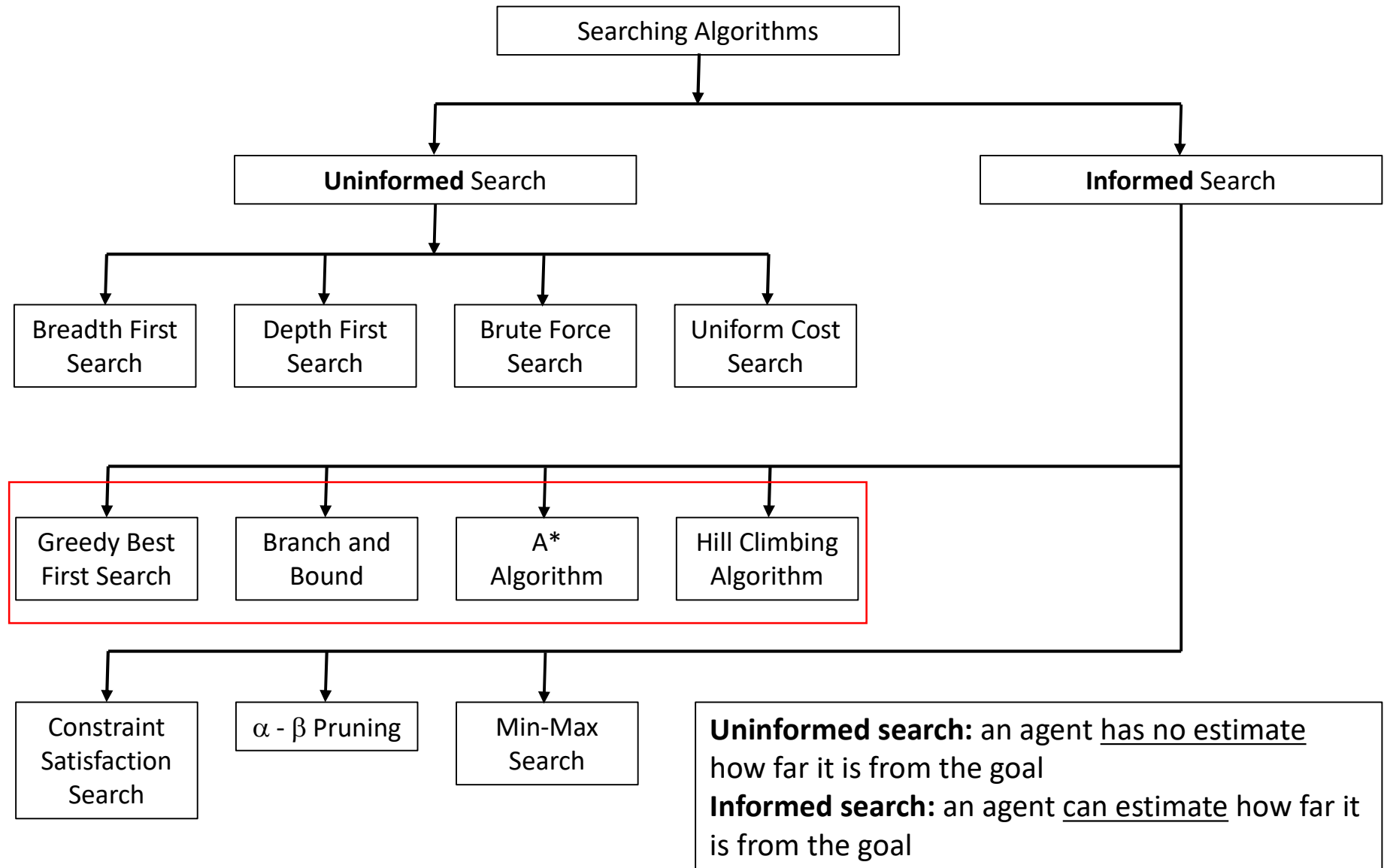
Search Tree: “Take Best First”



Expansion:

- Evaluation function $f(n) = \min$
- Use $\text{ACTIONS}(\mathbf{B}) = \{\text{toF}\}$ to get:
- $\text{RESULT}(\mathbf{B}, \text{toF}) = \mathbf{F}$

Selected Searching Algorithms



Informed Search: the Idea

When traversing the search tree **use domain knowledge / heuristics to avoid search paths that are likely to be fruitless**

Informed Search and Heuristics

Informed search relies on **domain-specific knowledge / hints** that help locate the goal state.

$$h(n) = h(\text{State } n)$$

$$h(n) = n(\text{relevant information about State } n)$$

$h(n)$: heuristic function - estimated cost of the cheapest path from State n to the goal state

Evaluation function

Calculate / obtain:

$$f(n) = f(\text{State } n)$$

$$f(n) = f(\text{relevant information about State } n)$$

**A state n with minimum (or maximum) $f(n)$
should be chosen for expansion**

What about ties?

Best-First Search

```
function BEST-FIRST-SEARCH(problem, f) returns a solution node or failure  
  node  $\leftarrow$  NODE(STATE=problem.INITIAL)  
  frontier  $\leftarrow$  a priority queue ordered by f, with node as an element  
  reached  $\leftarrow$  a lookup table, with one entry with key problem.INITIAL and value node  
  while not IS-EMPTY(frontier) do  
    node  $\leftarrow$  POP(frontier)  
    if problem.IS-GOAL(node.STATE) then return node  
    for each child in EXPAND(problem, node) do  
      s  $\leftarrow$  child.STATE  
      if s is not in reached or child.PATH-COST < reached[s].PATH-COST then  
        reached[s]  $\leftarrow$  child  
        add child to frontier  
  return failure
```

Best-First Search is really a class of search algorithms that:

- Use the **evaluation function $f(n)$** to pick next action
- Keep track of **visited states**
- Keep track of **frontier states**
- **Evaluation function $f(n)$ choice controls their behavior**

Informed Search and Heuristics

Informed search relies on **domain-specific knowledge / hints** that help locate the goal state.

$$h(n) = h(\text{State } n)$$

$$h(n) = n(\text{relevant information about State } n)$$

$h(n)$: heuristic function - estimated cost of the cheapest path from State n to the goal state

Hill Climbing Search

- **The most primitive informed search approach**
 - a naive greedy algorithm
 - evaluation function: the cost of next move
 - does not care about the “bigger picture” (for example: total search path cost)
- **Practicalities:**
 - usually does not keep track of search history:
 - not tracking visited nodes → loops!
 - not tracking frontier nodes → does not look at alternatives

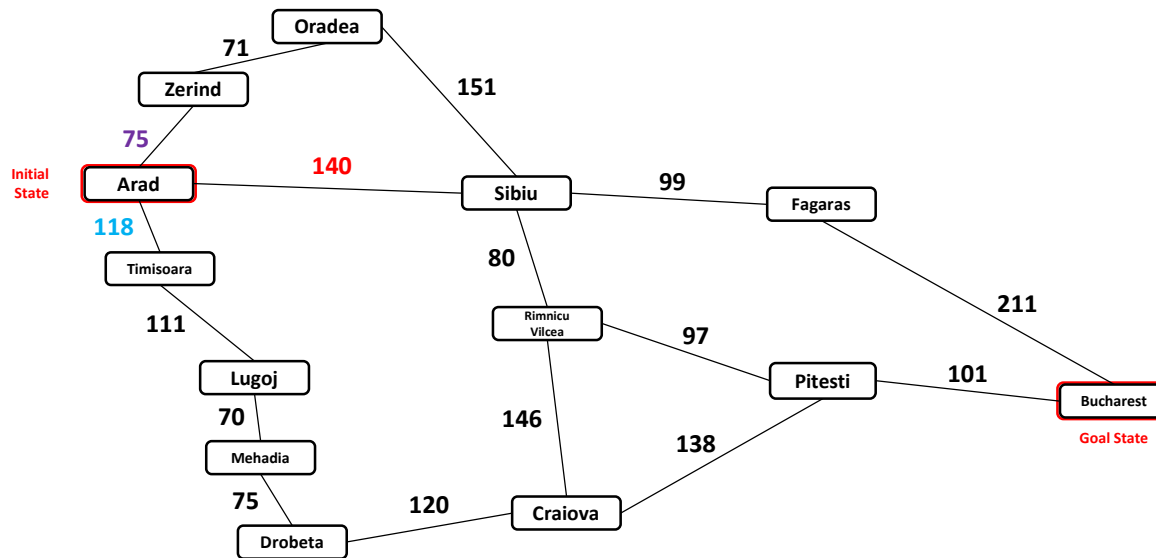
Hill Climbing: Evaluation function

Calculate / obtain:

$$f(n) = \text{ACTION-COST}(\text{State}_a, \text{toState}_n, \text{State}_n)$$

A state n with minimum (or maximum) $f(n)$
should be chosen for expansion

Dracula's Roadtrip: Hill Climbing

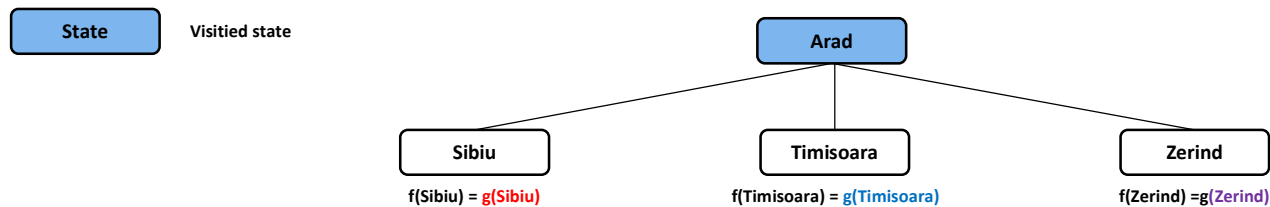
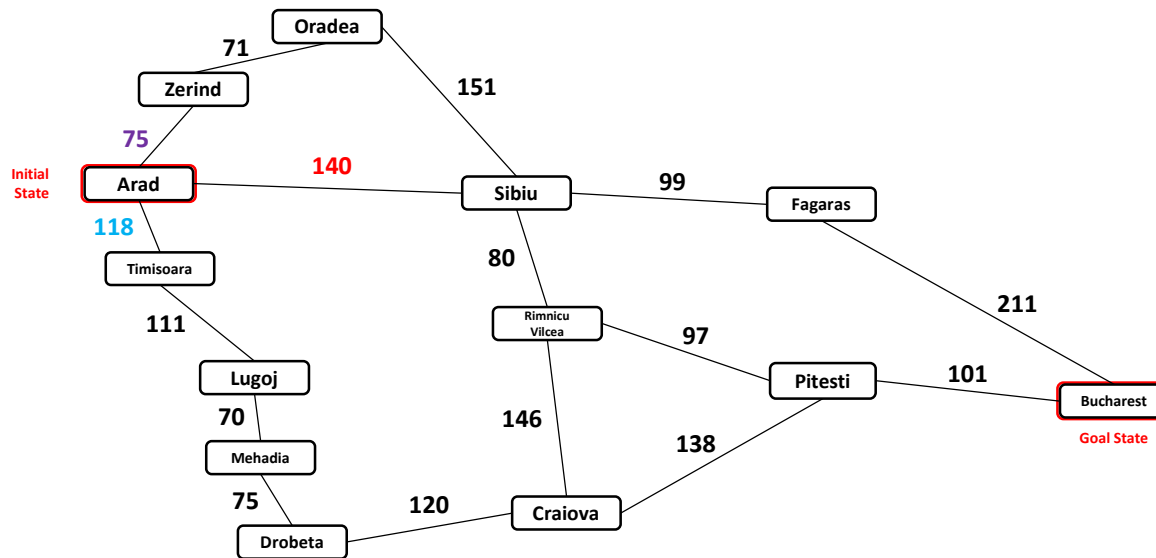


State

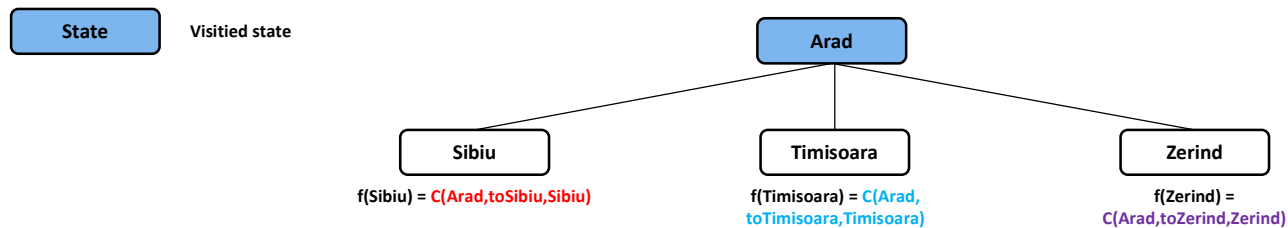
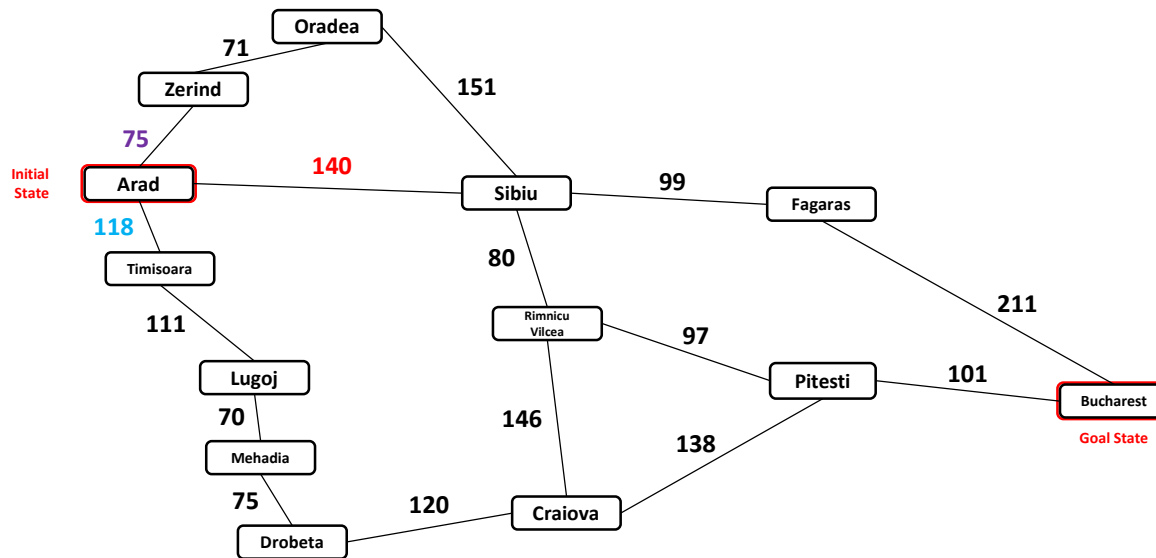
Visited state

Arad

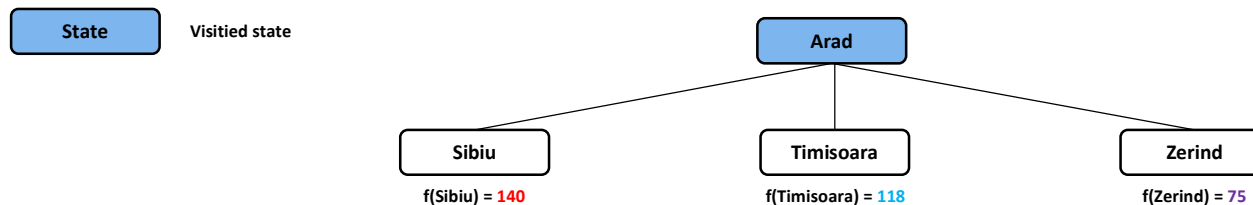
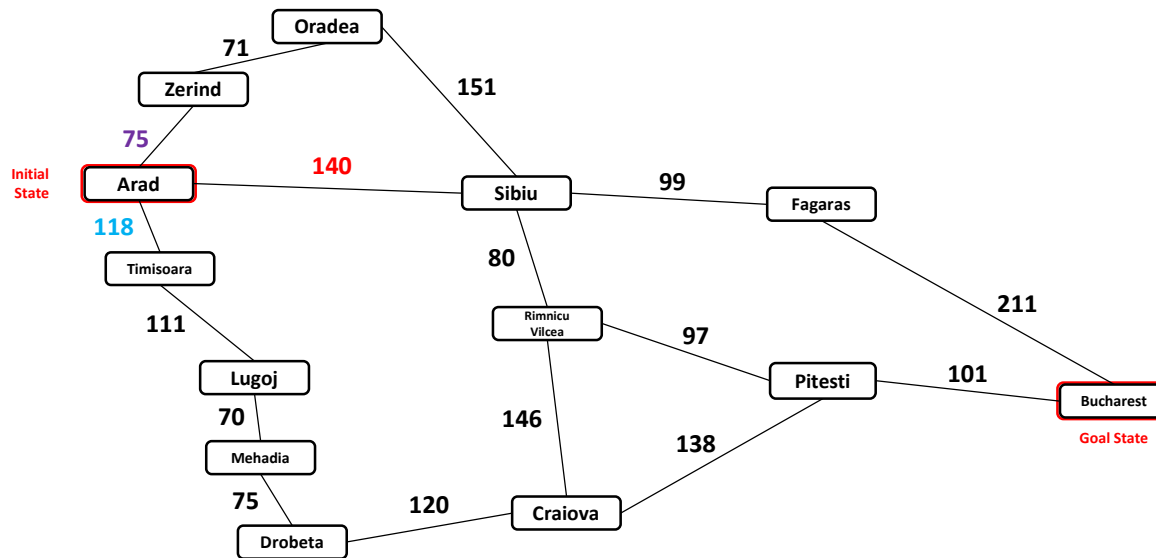
Dracula's Roadtrip: Hill Climbing



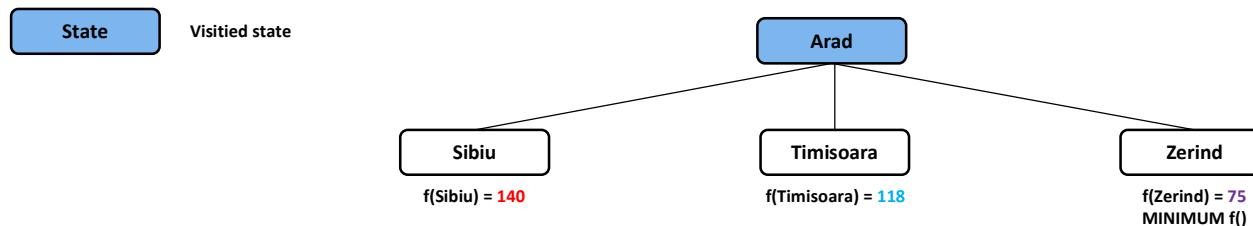
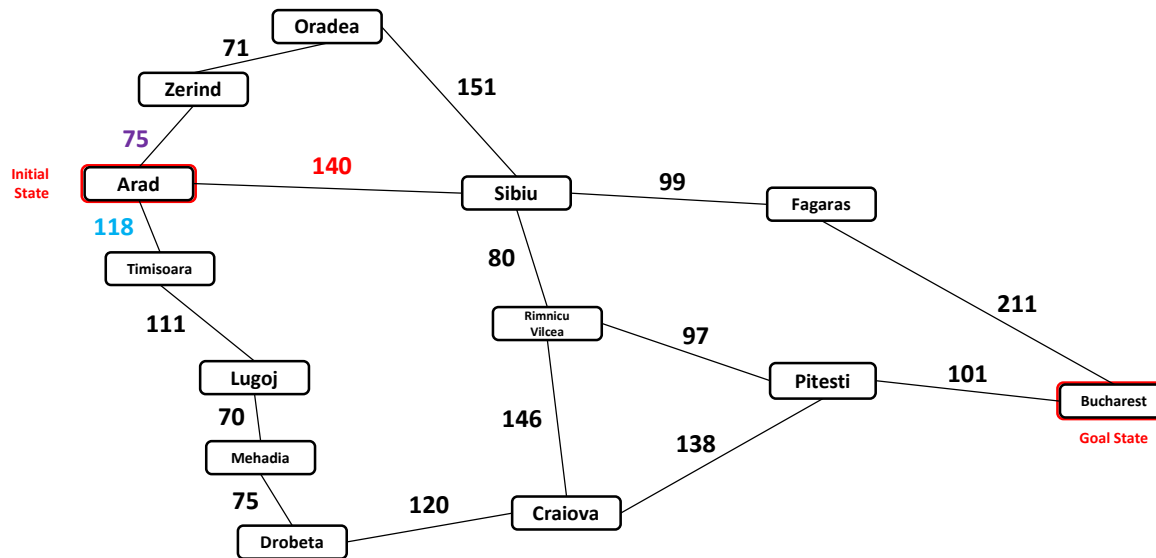
Dracula's Roadtrip: Hill Climbing



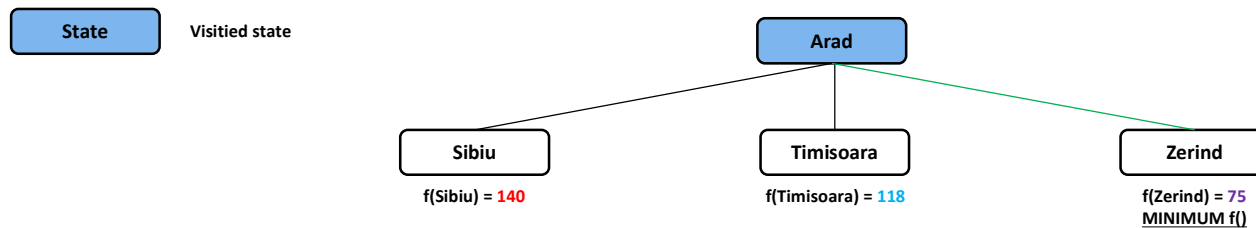
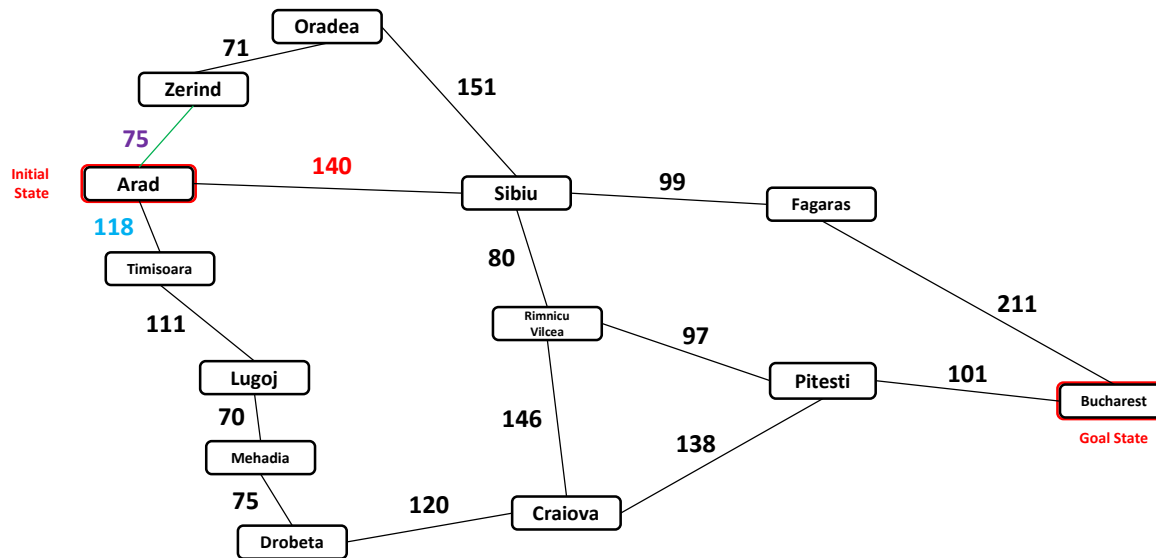
Dracula's Roadtrip: Hill Climbing



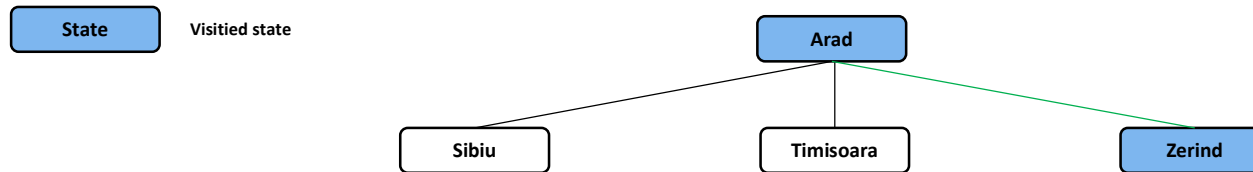
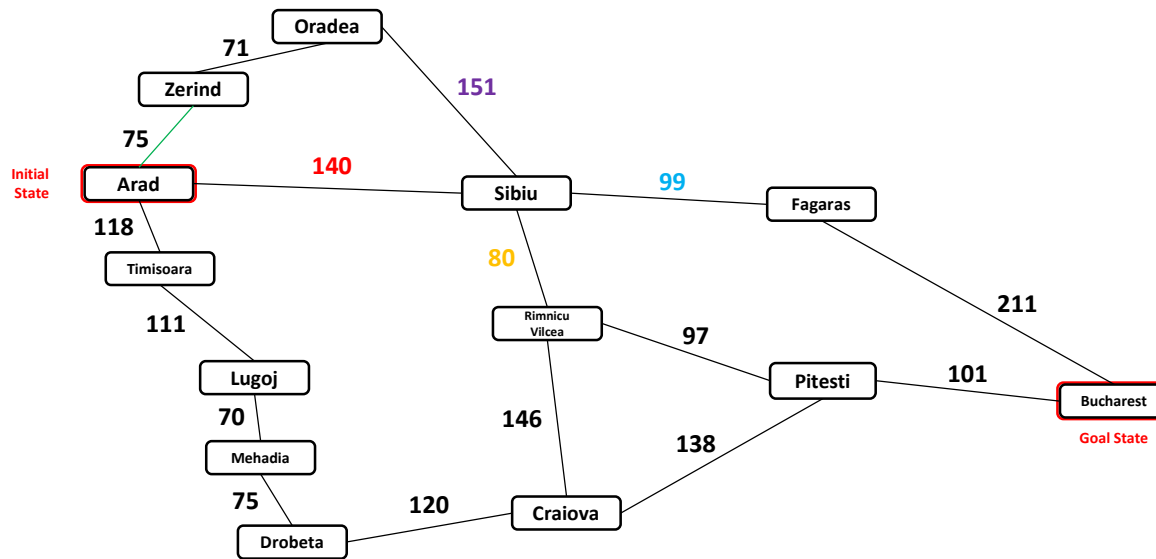
Dracula's Roadtrip: Hill Climbing



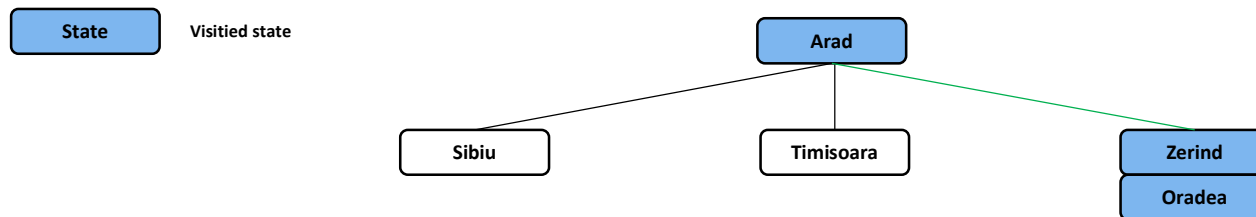
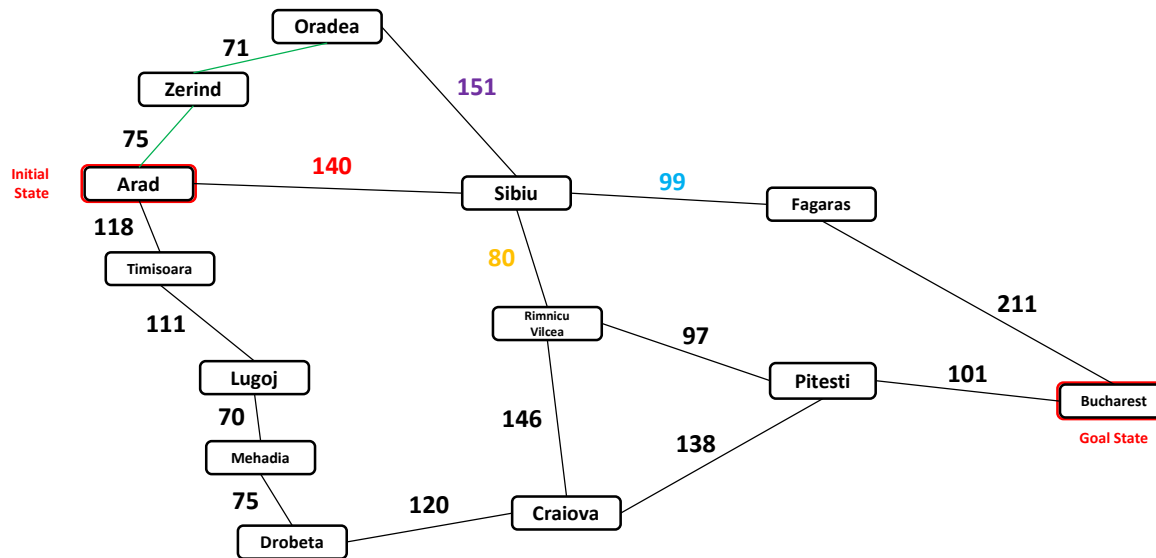
Dracula's Roadtrip: Hill Climbing



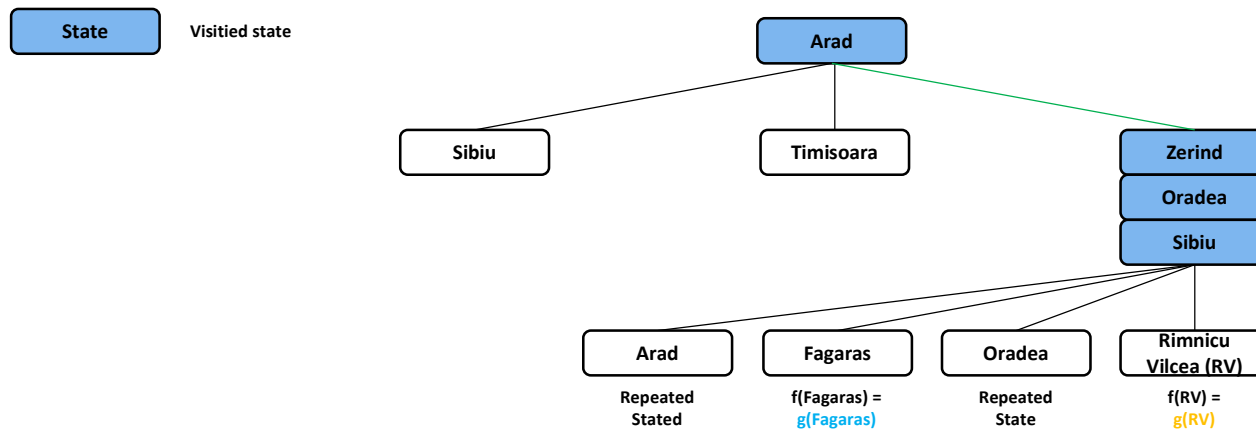
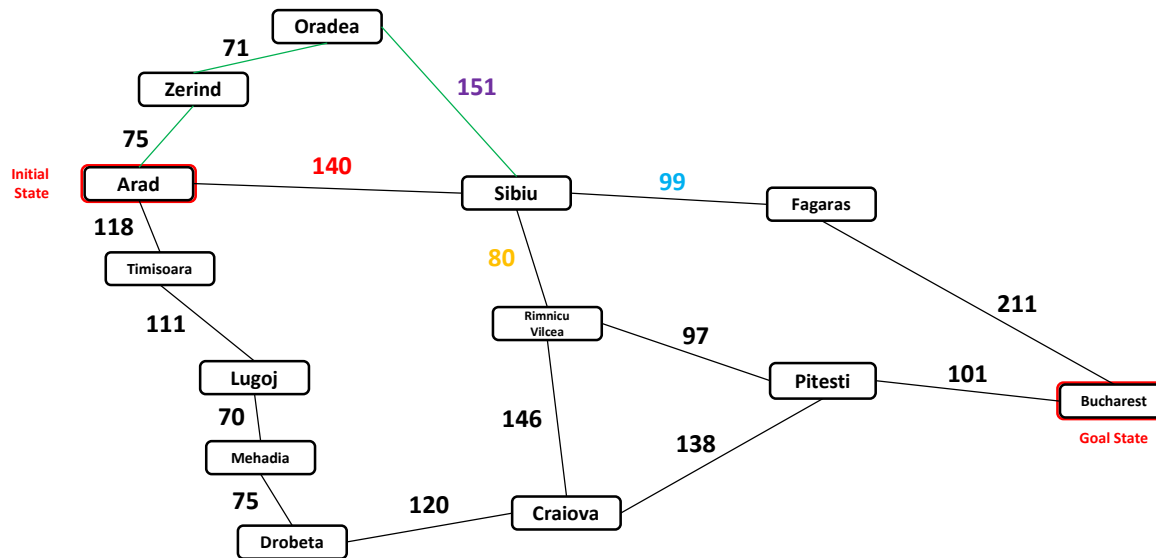
Dracula's Roadtrip: Hill Climbing



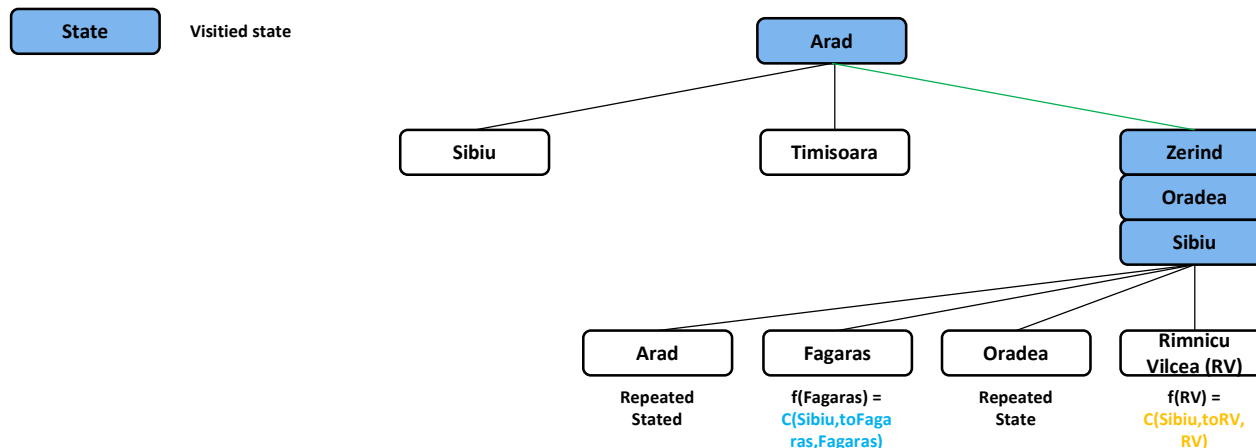
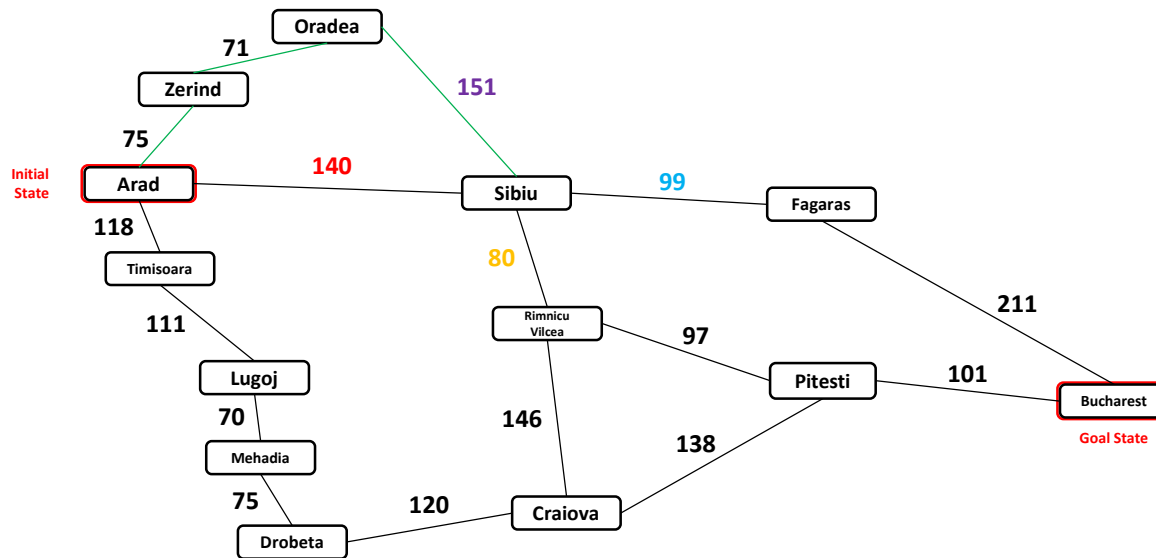
Dracula's Roadtrip: Hill Climbing



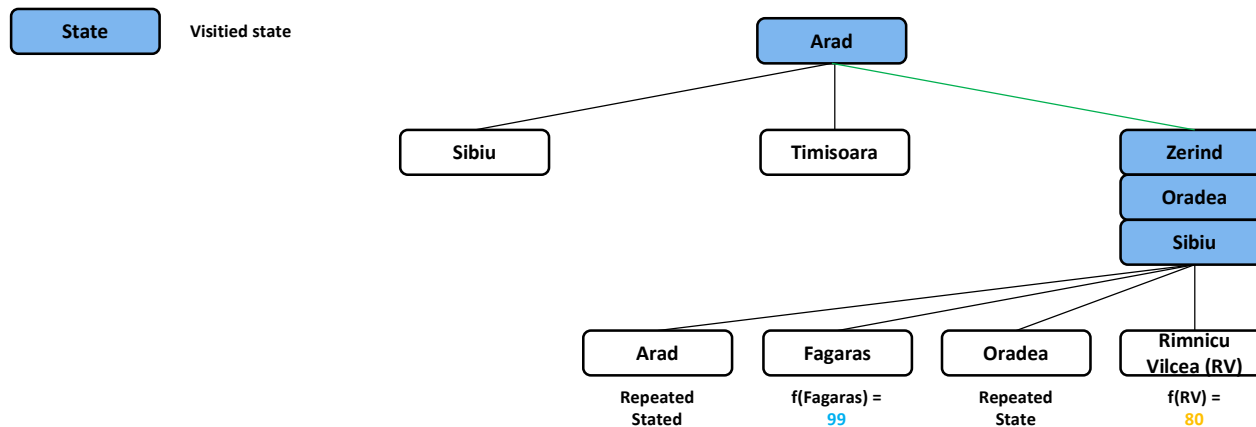
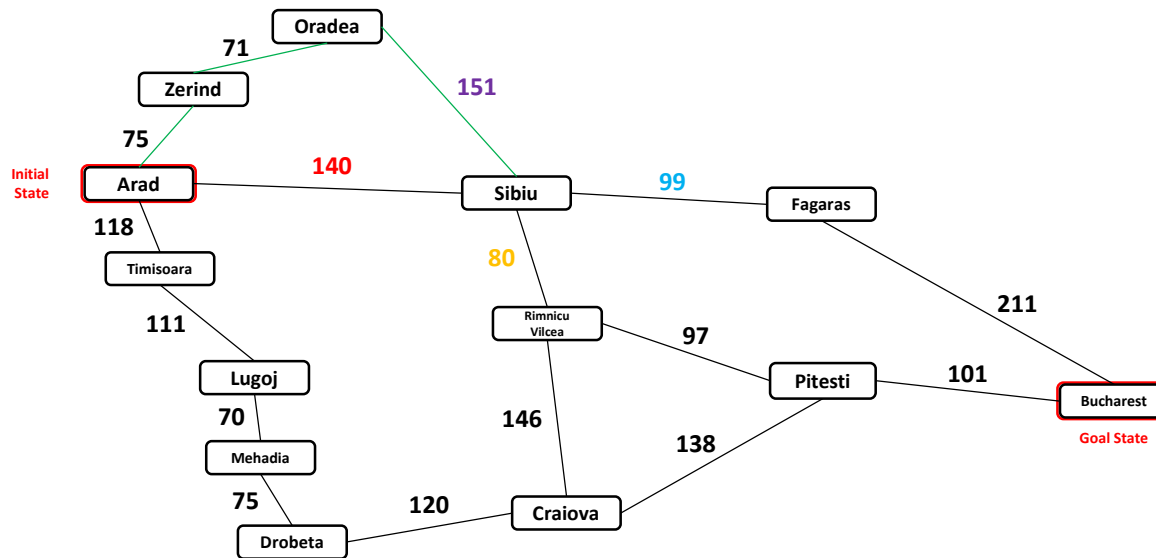
Dracula's Roadtrip: Hill Climbing



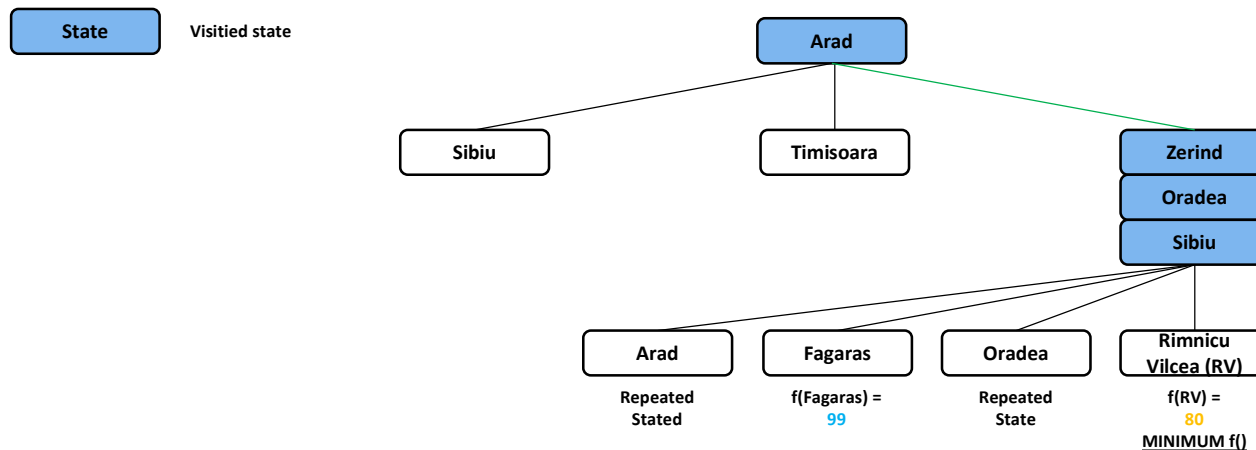
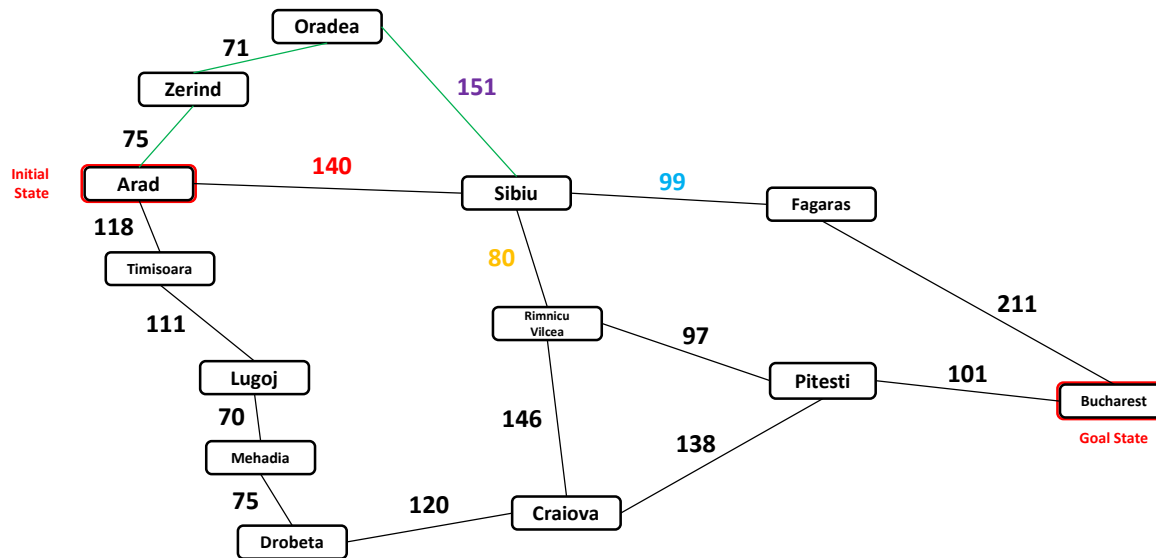
Dracula's Roadtrip: Hill Climbing



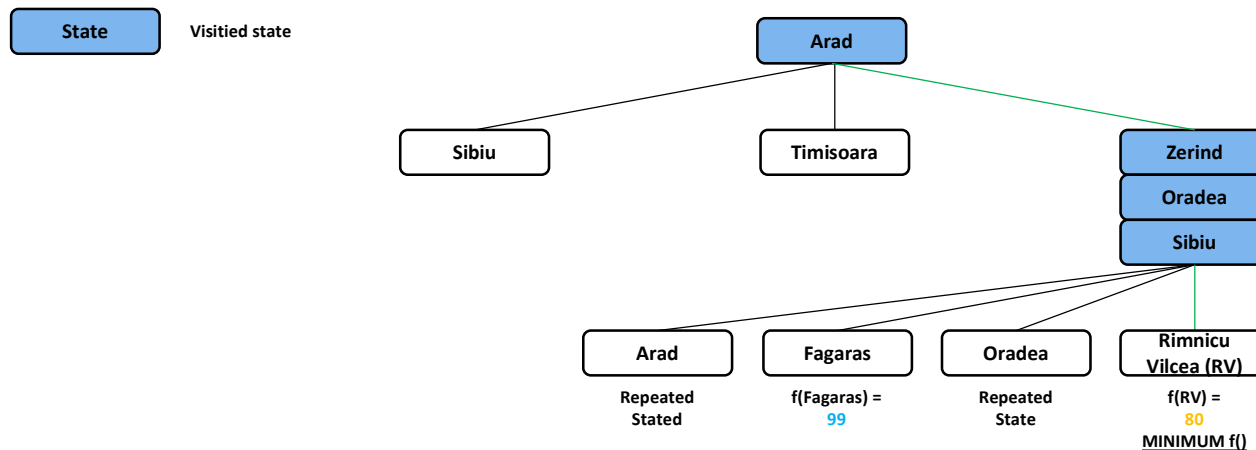
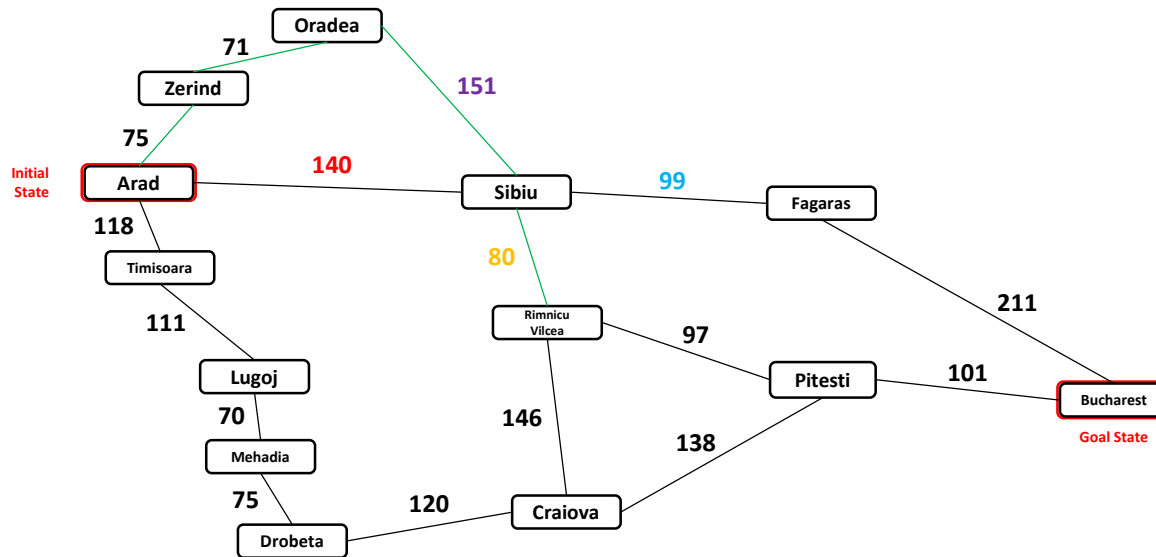
Dracula's Roadtrip: Hill Climbing



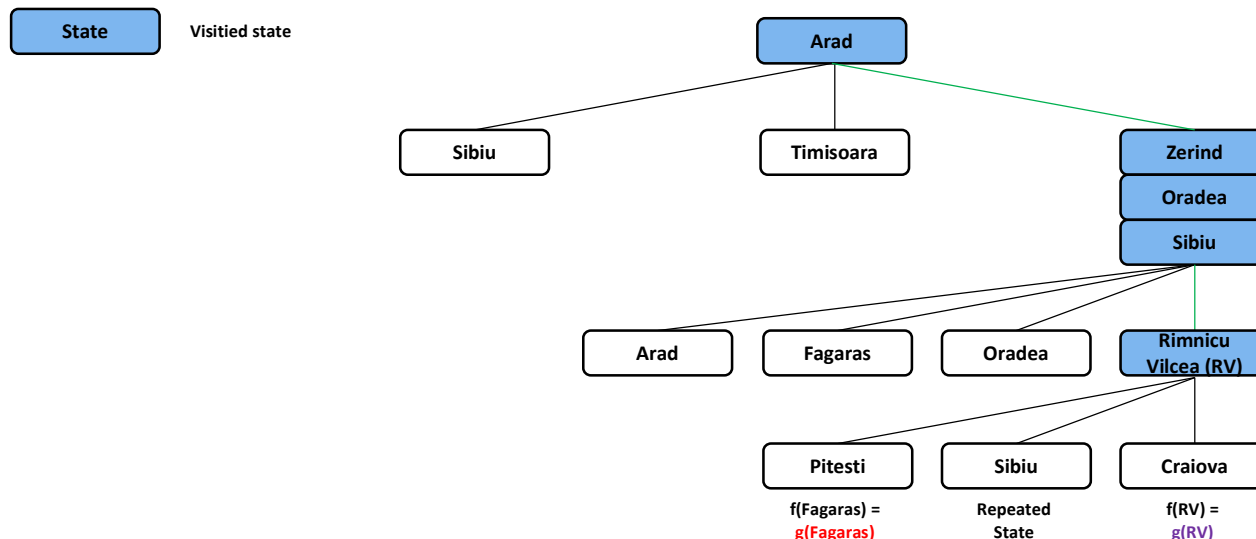
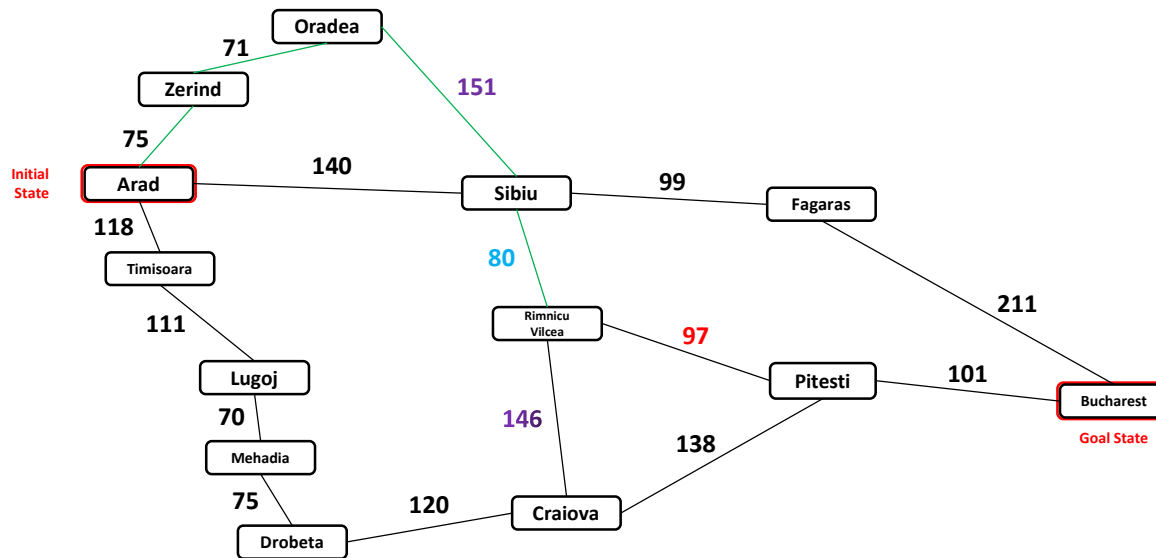
Dracula's Roadtrip: Hill Climbing



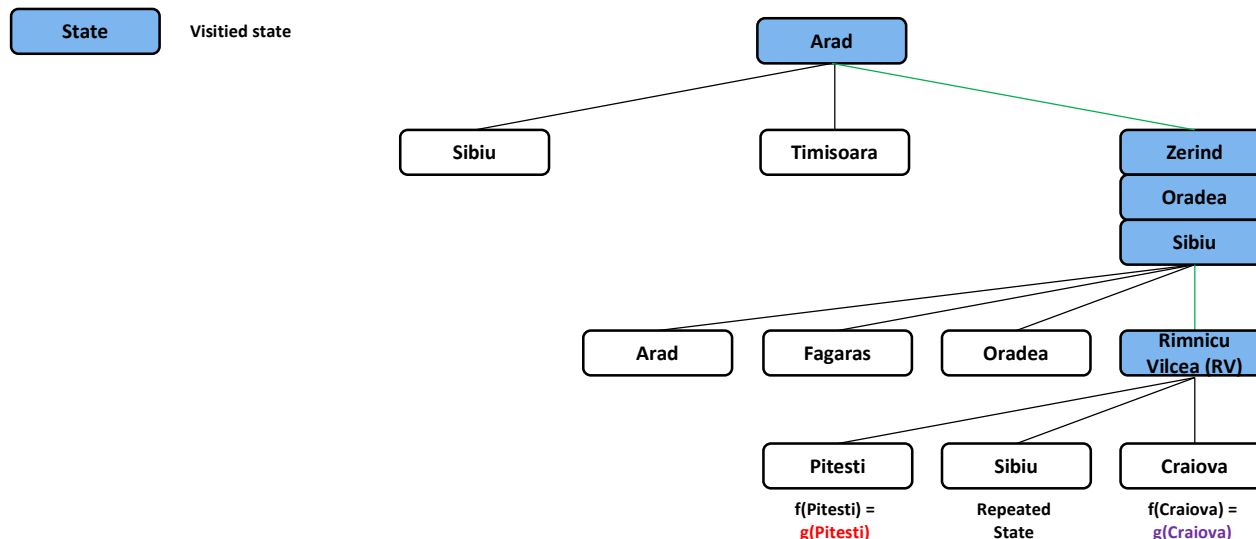
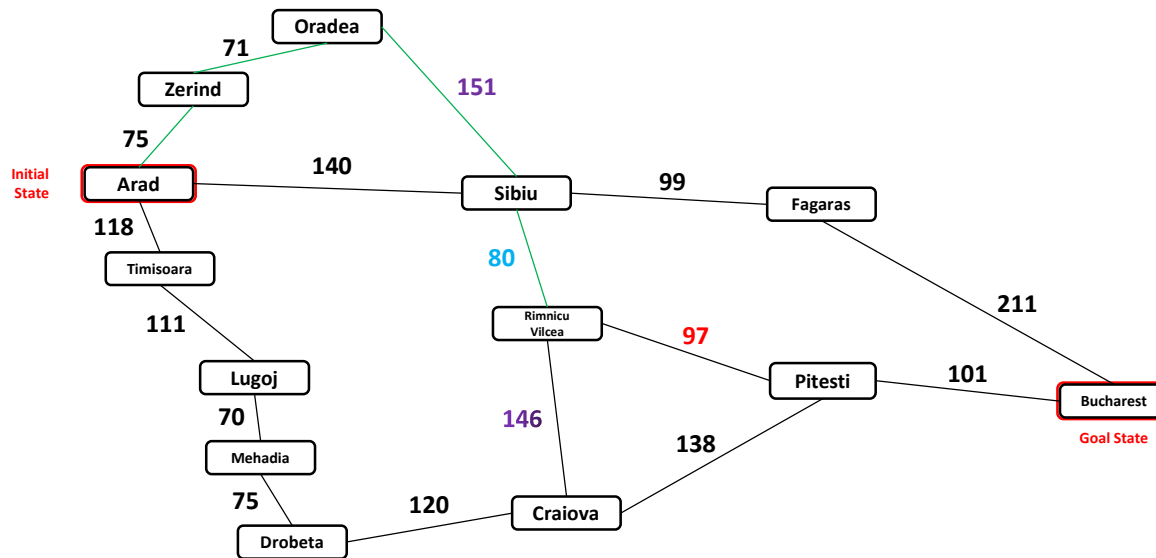
Dracula's Roadtrip: Hill Climbing



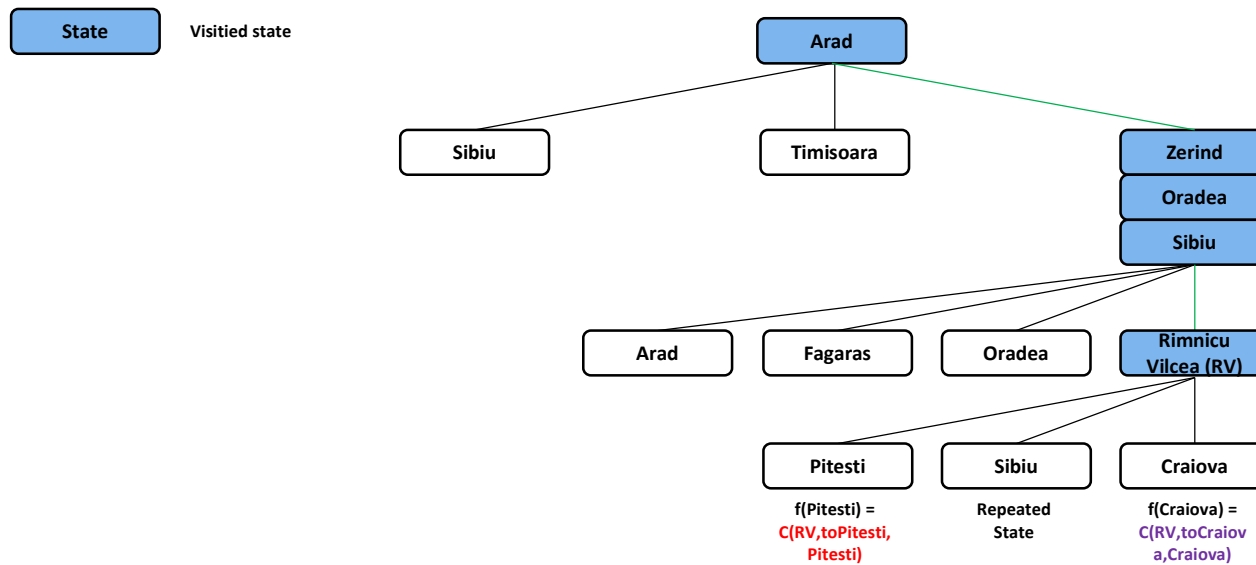
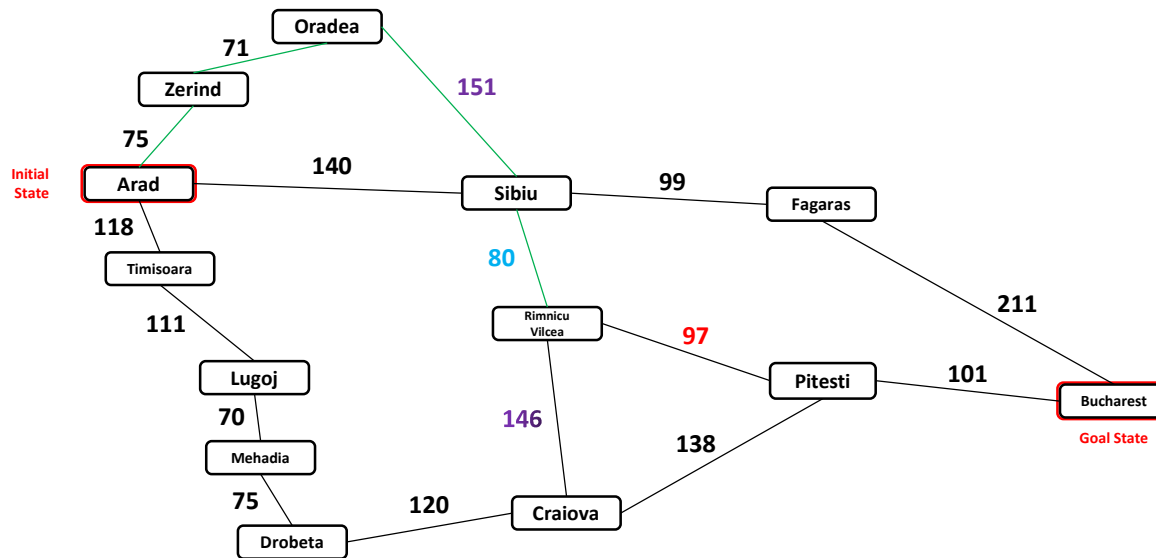
Dracula's Roadtrip: Hill Climbing



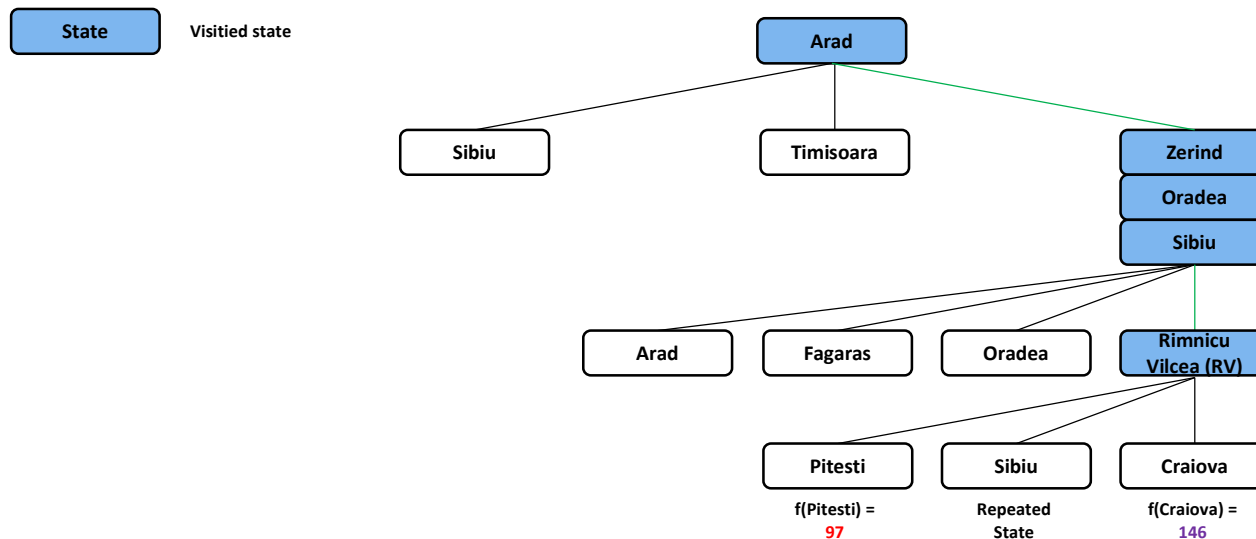
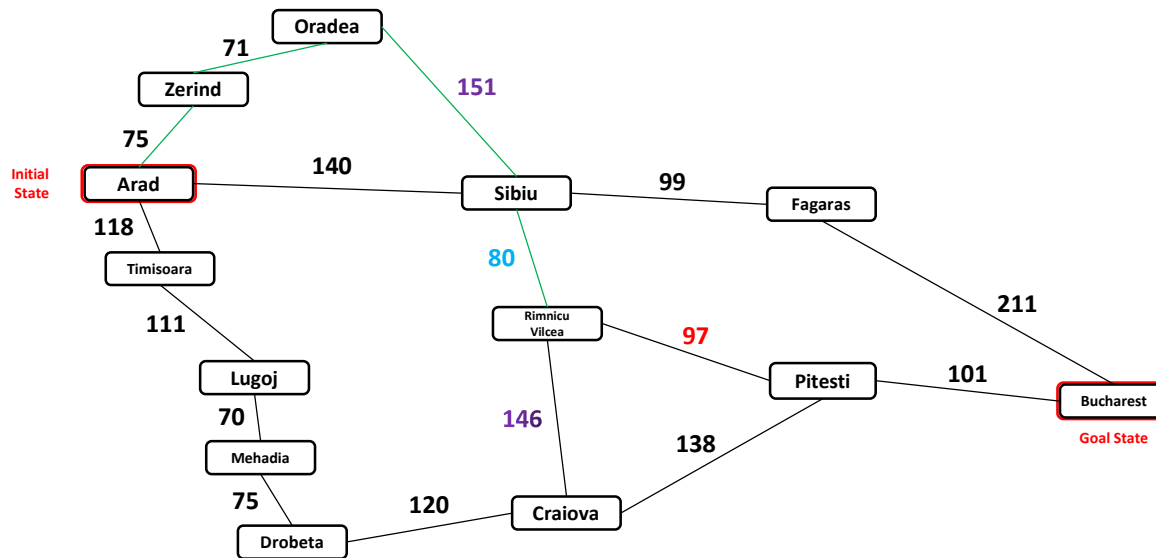
Dracula's Roadtrip: Hill Climbing



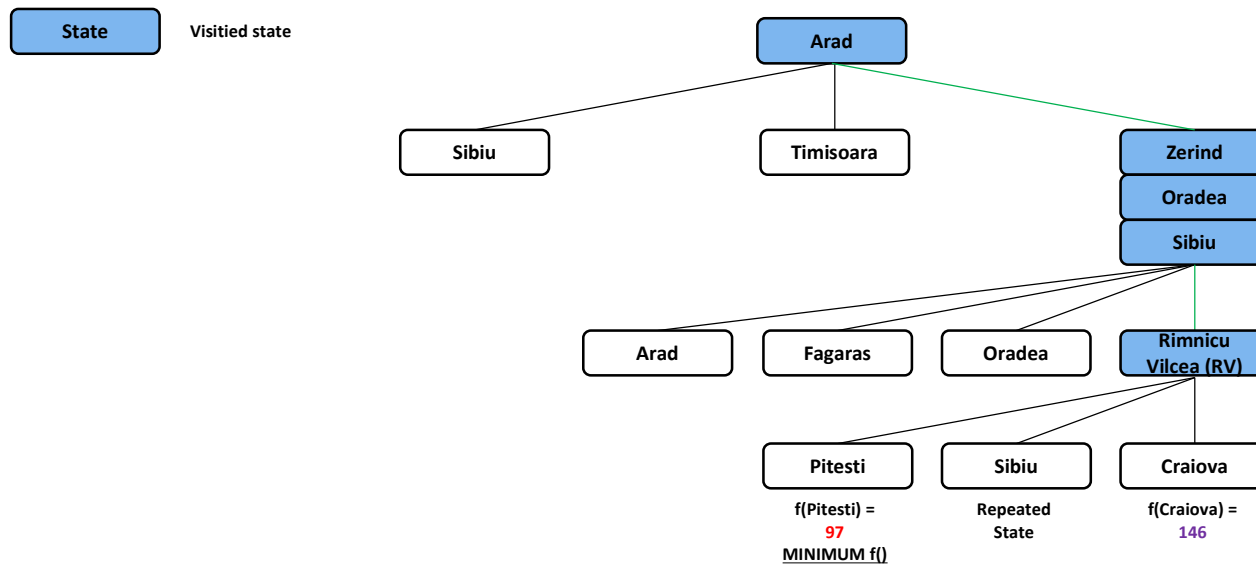
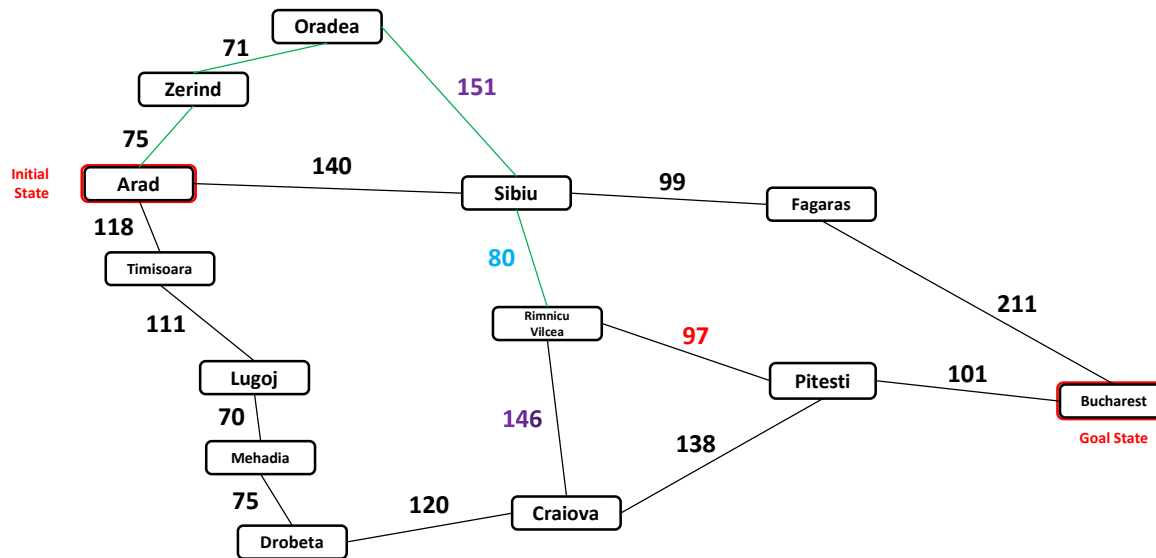
Dracula's Roadtrip: Hill Climbing



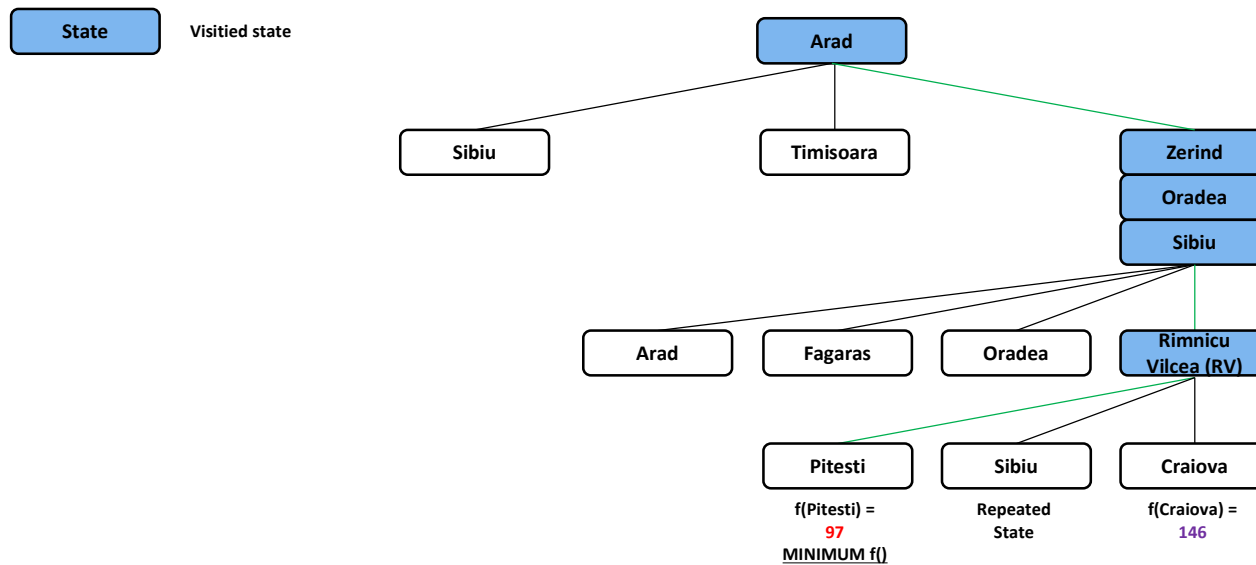
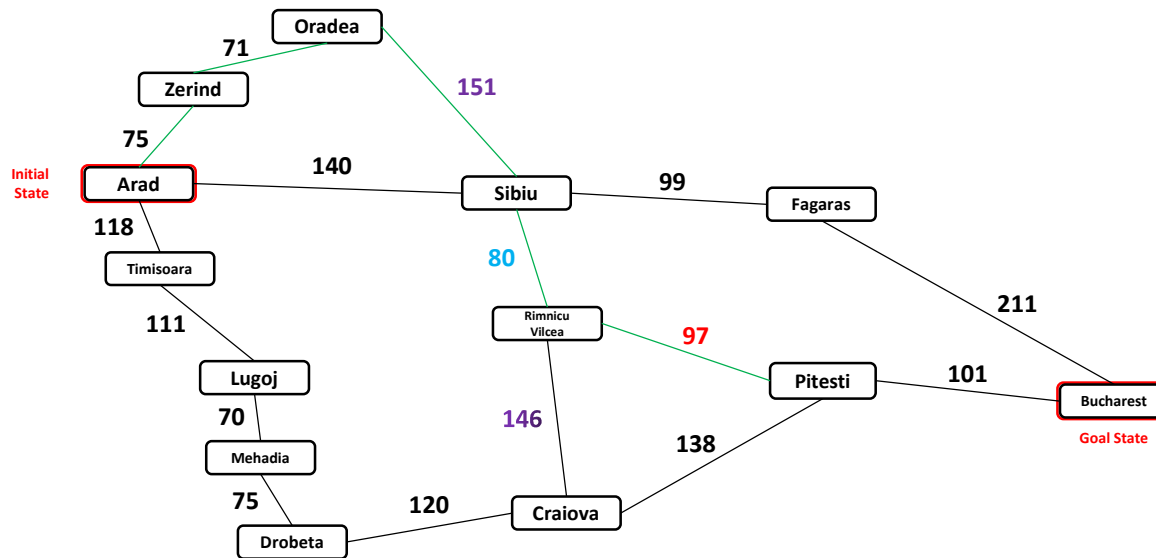
Dracula's Roadtrip: Hill Climbing



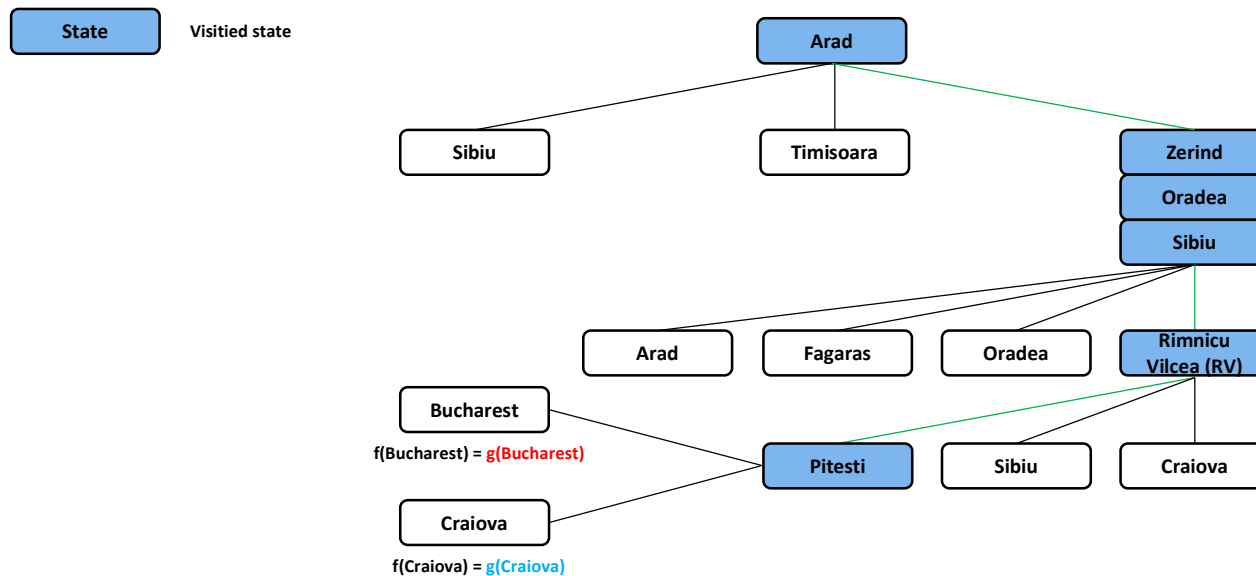
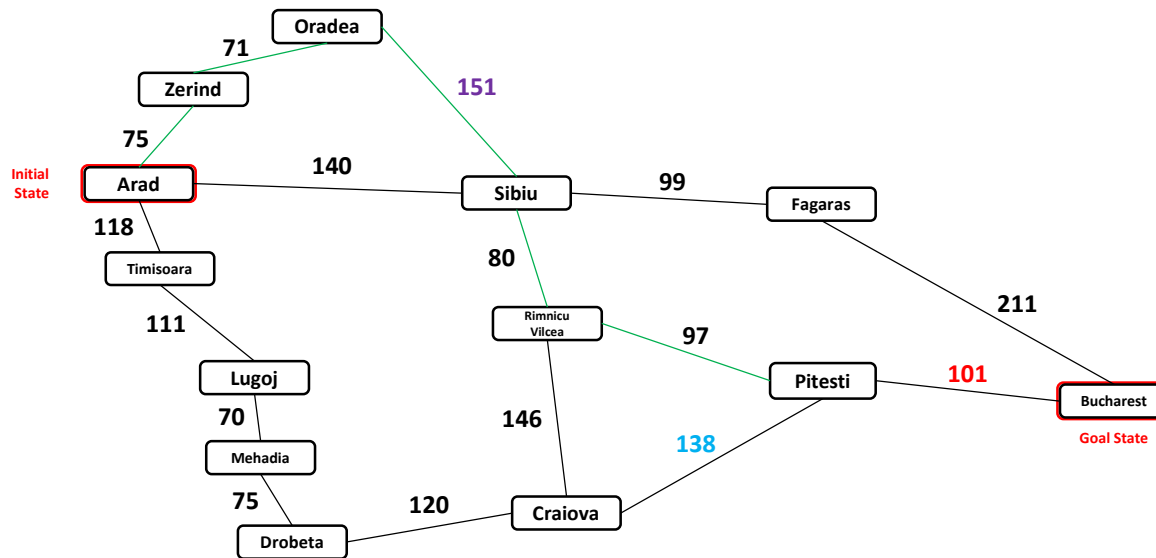
Dracula's Roadtrip: Hill Climbing



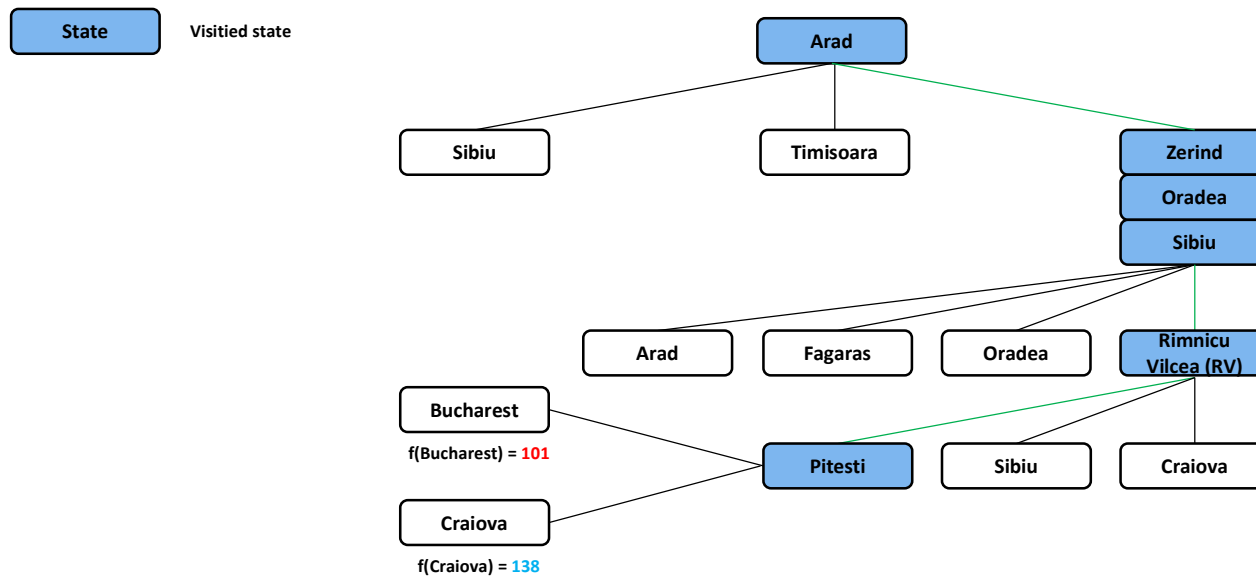
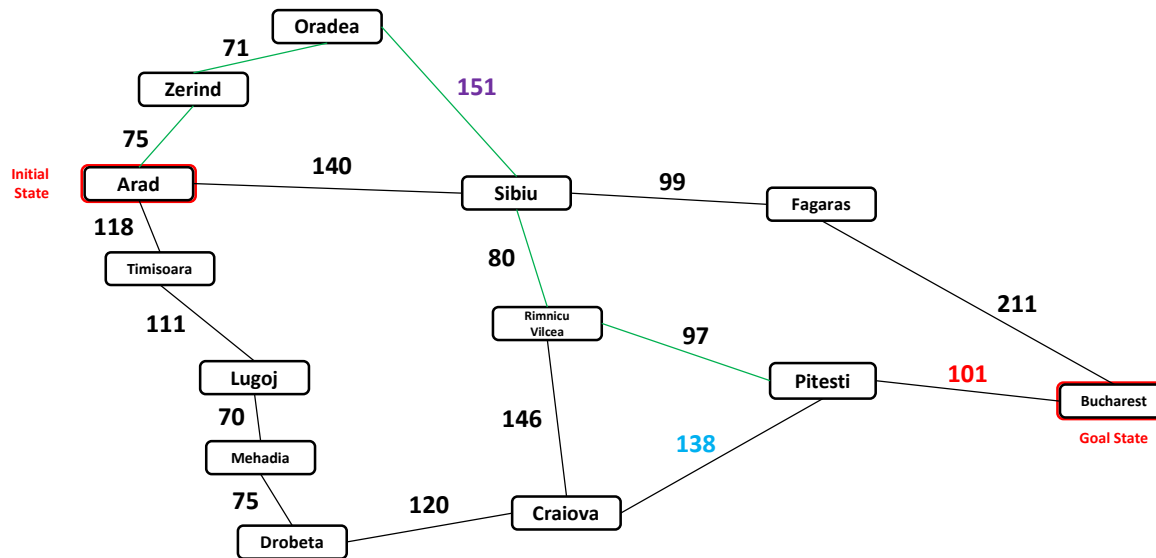
Dracula's Roadtrip: Hill Climbing



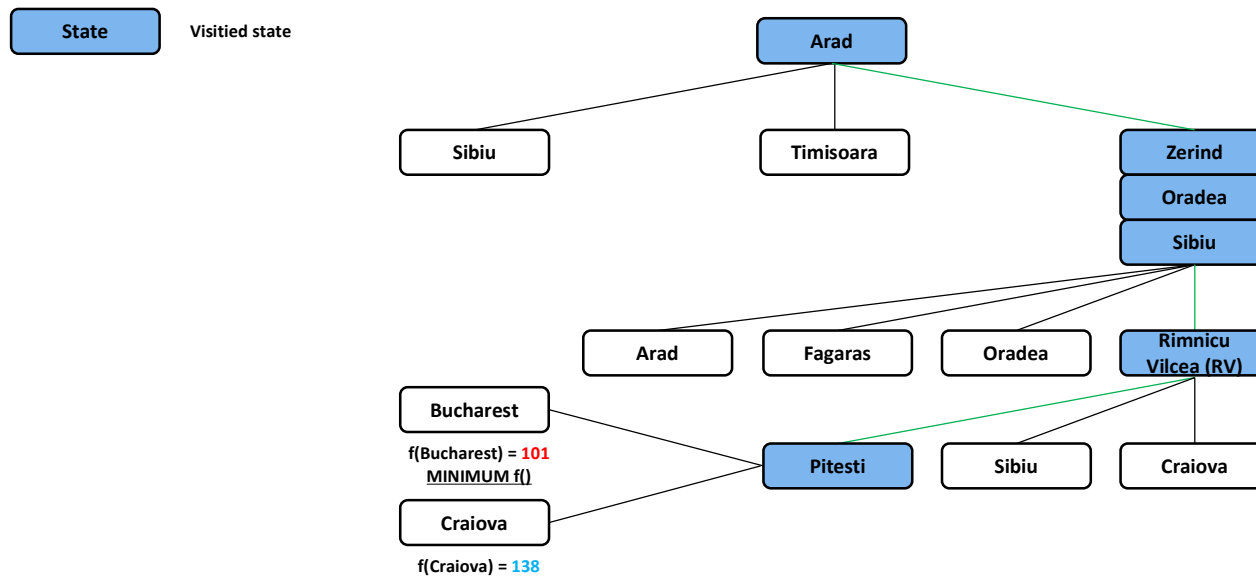
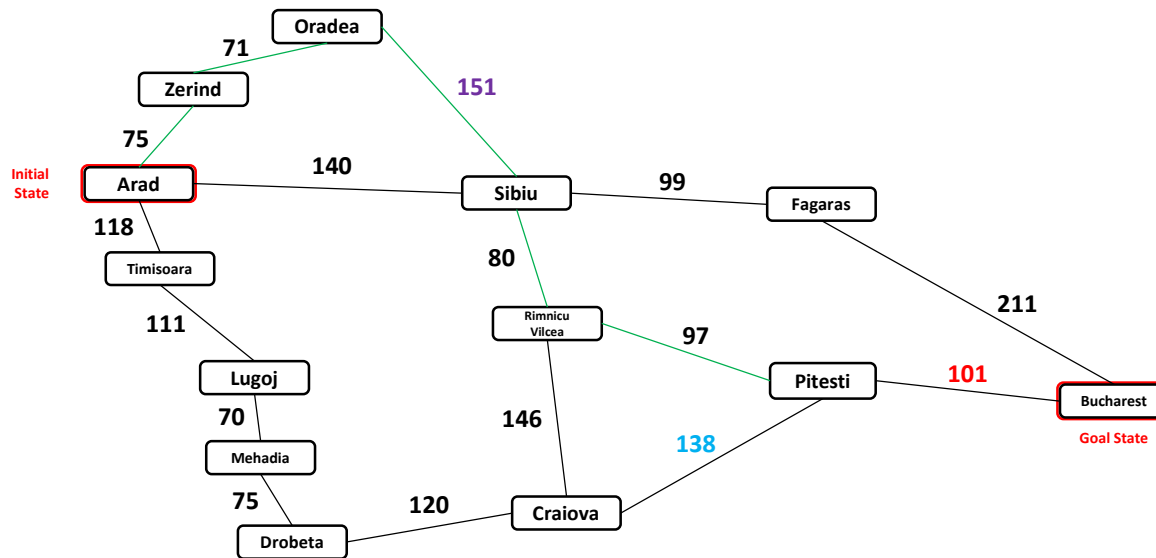
Dracula's Roadtrip: Hill Climbing



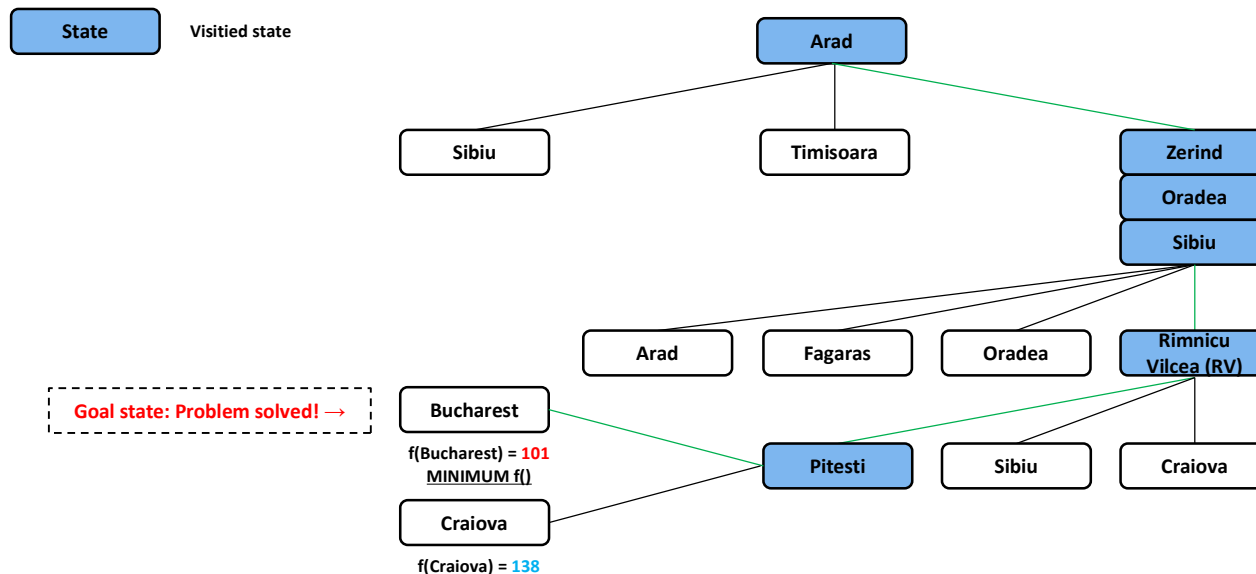
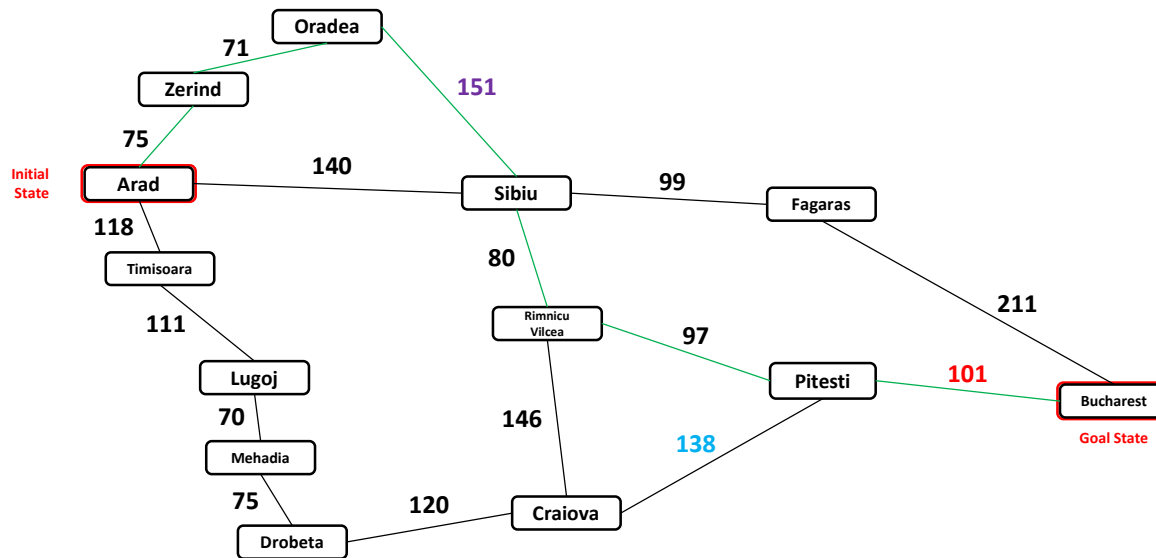
Dracula's Roadtrip: Hill Climbing



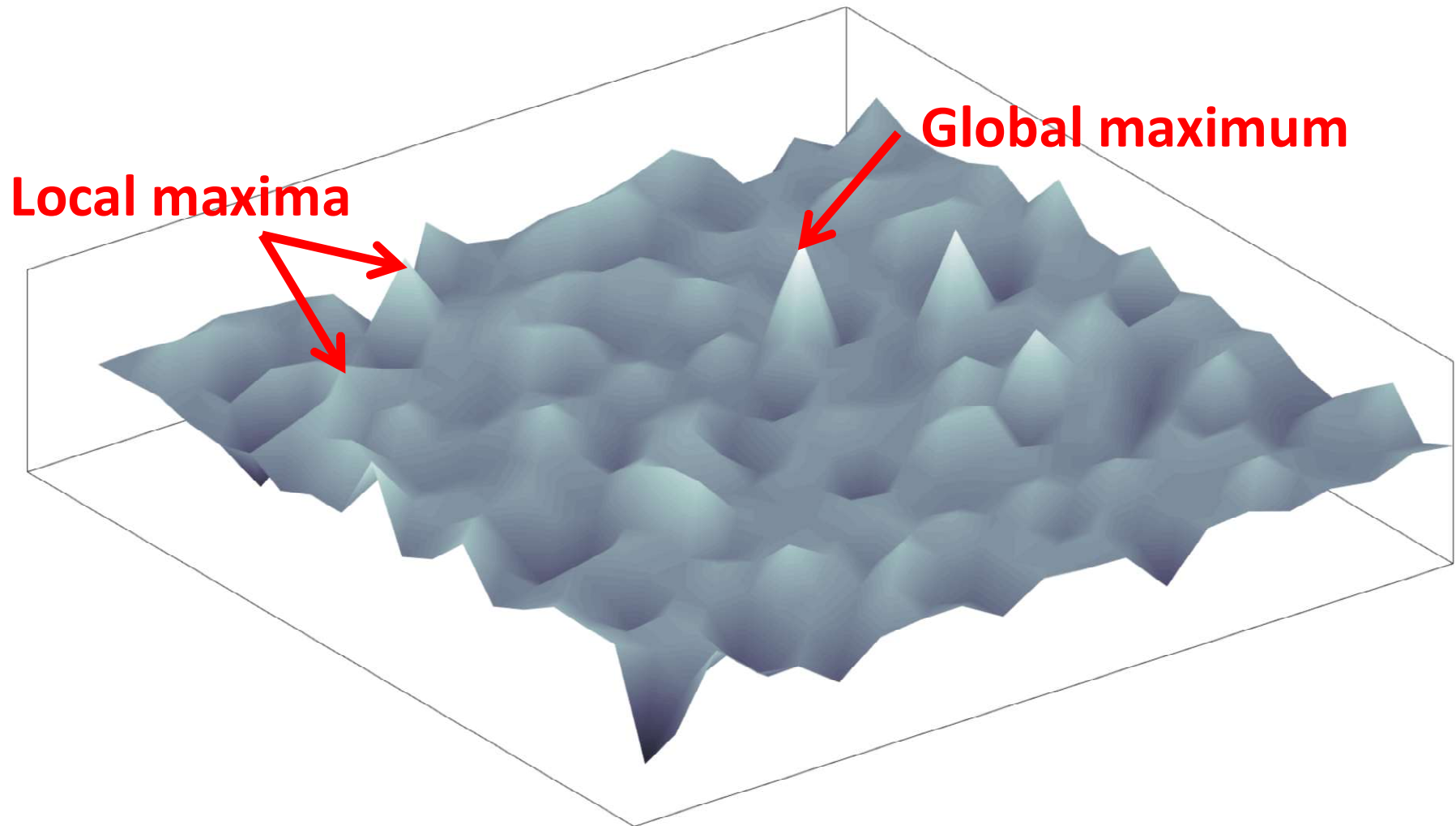
Dracula's Roadtrip: Hill Climbing



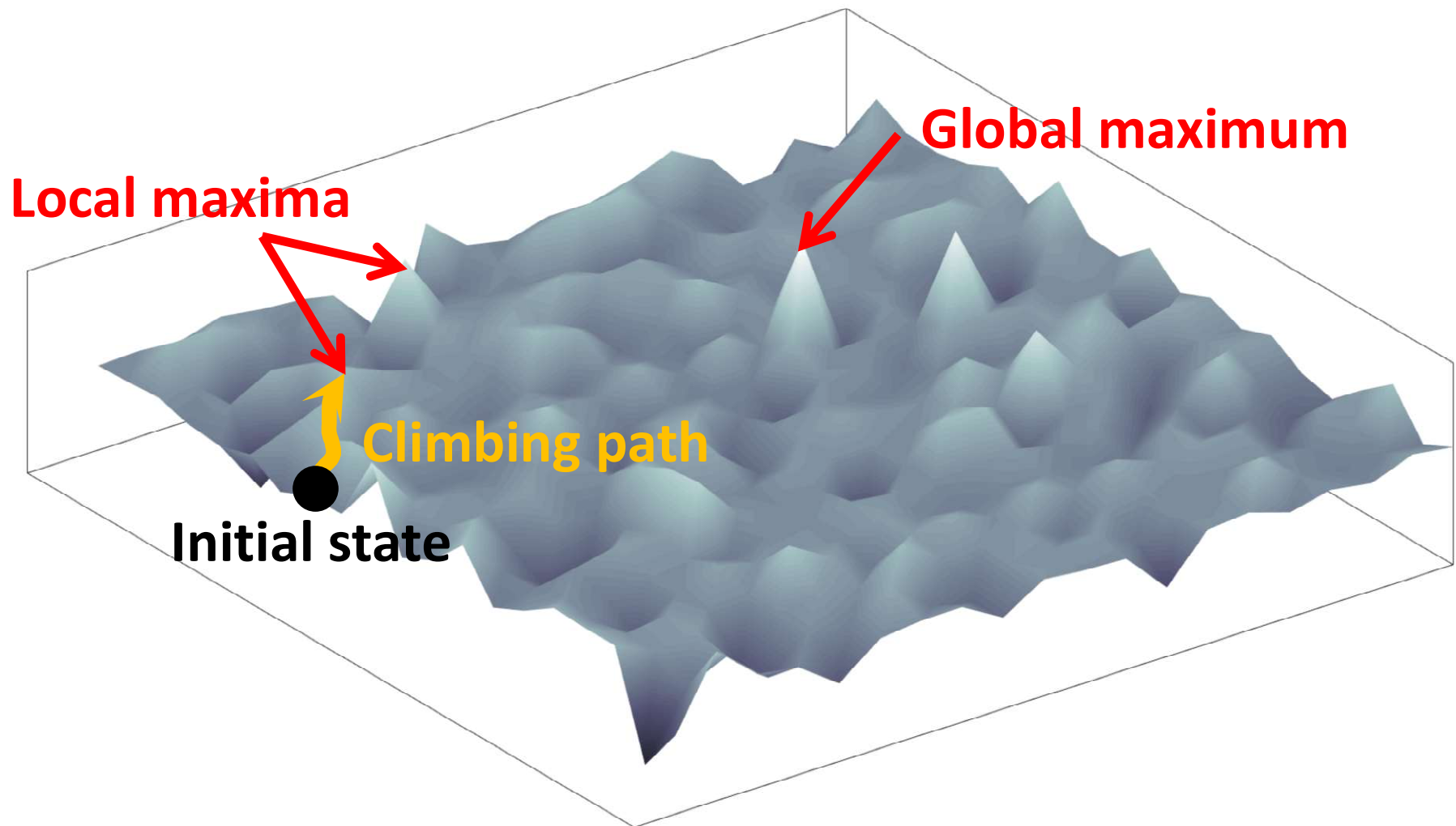
Dracula's Roadtrip: Hill Climbing



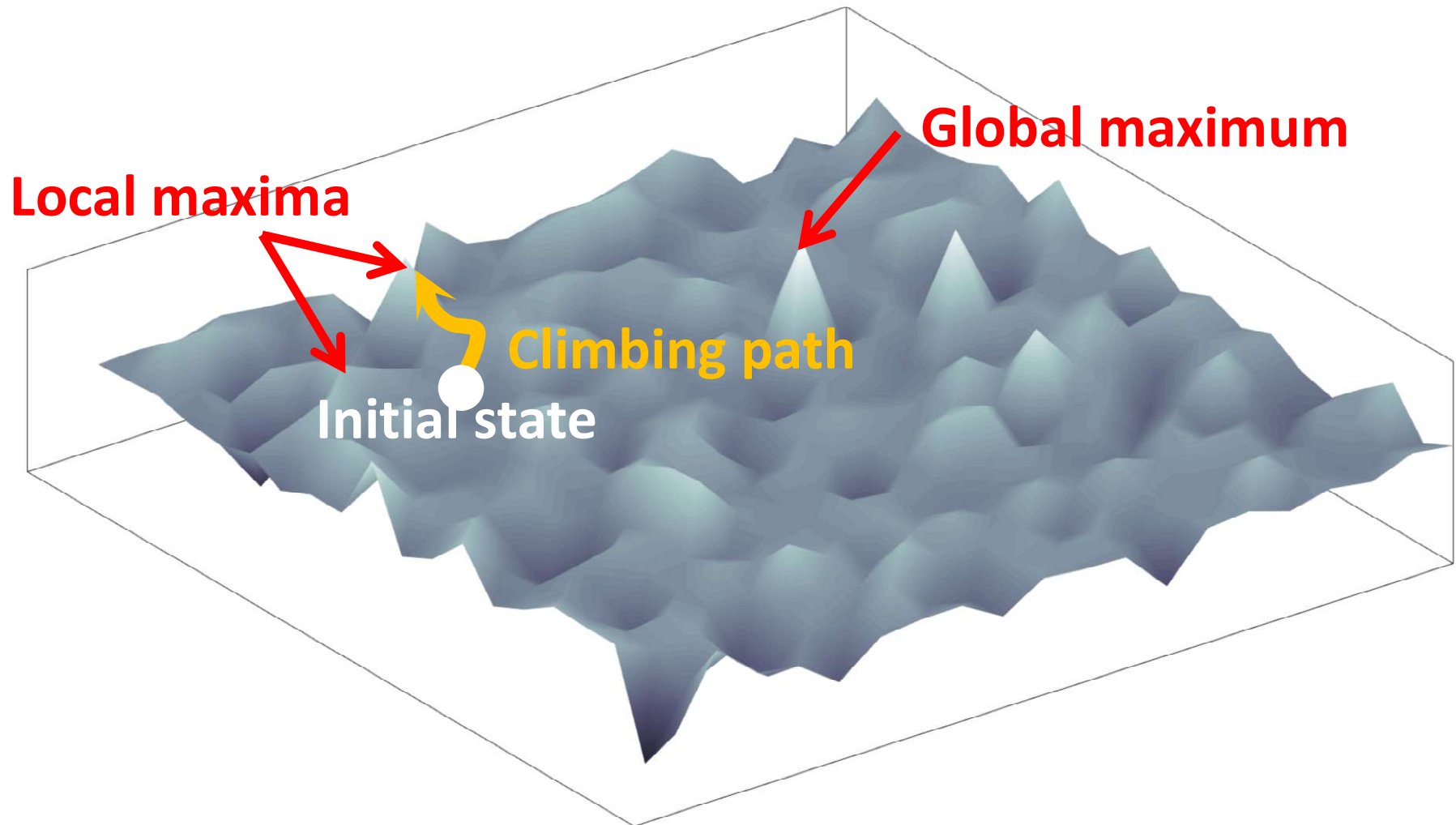
Hill Climbing Problems: Local Maxima



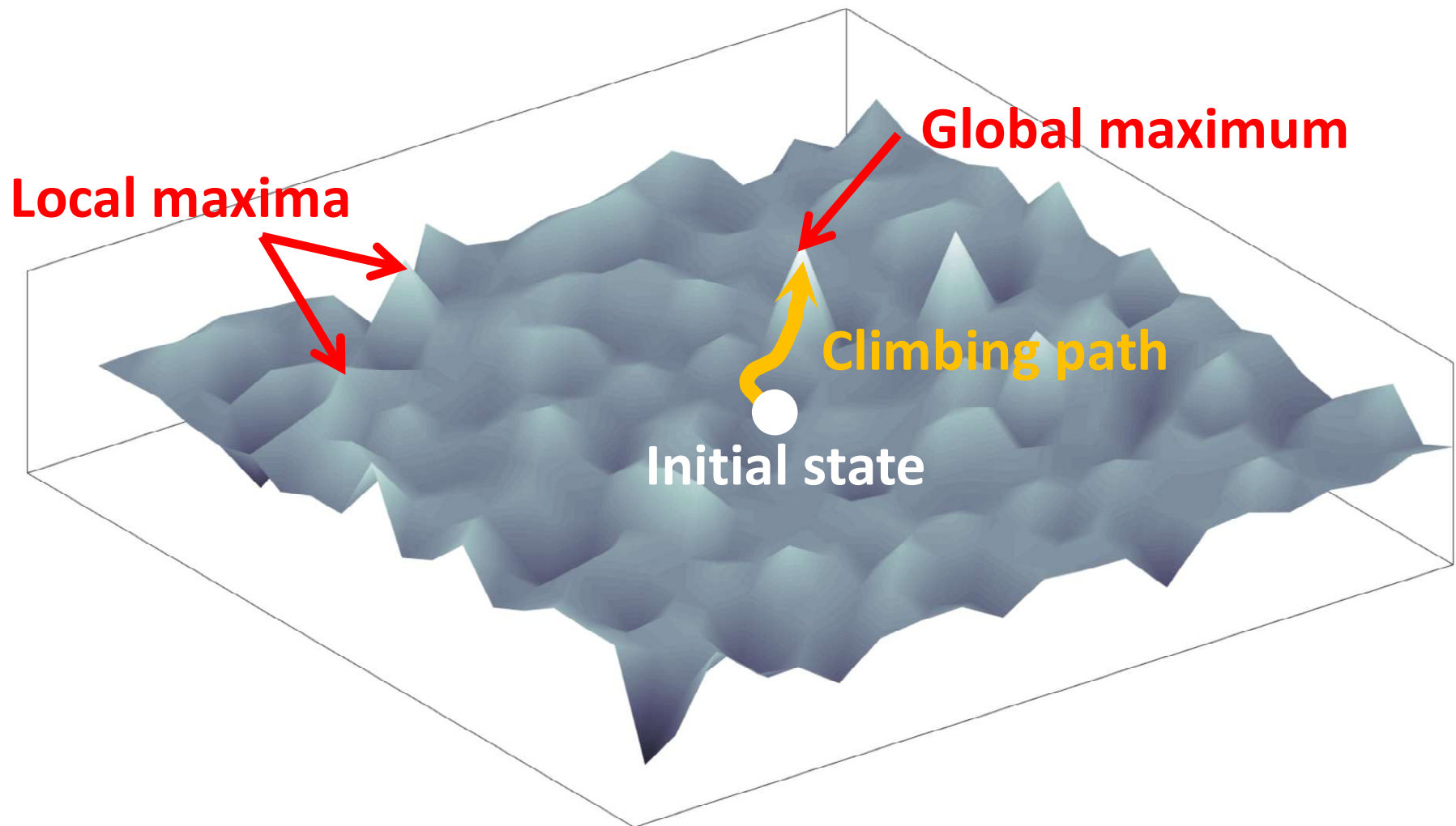
Hill Climbing Problems: Local Maxima



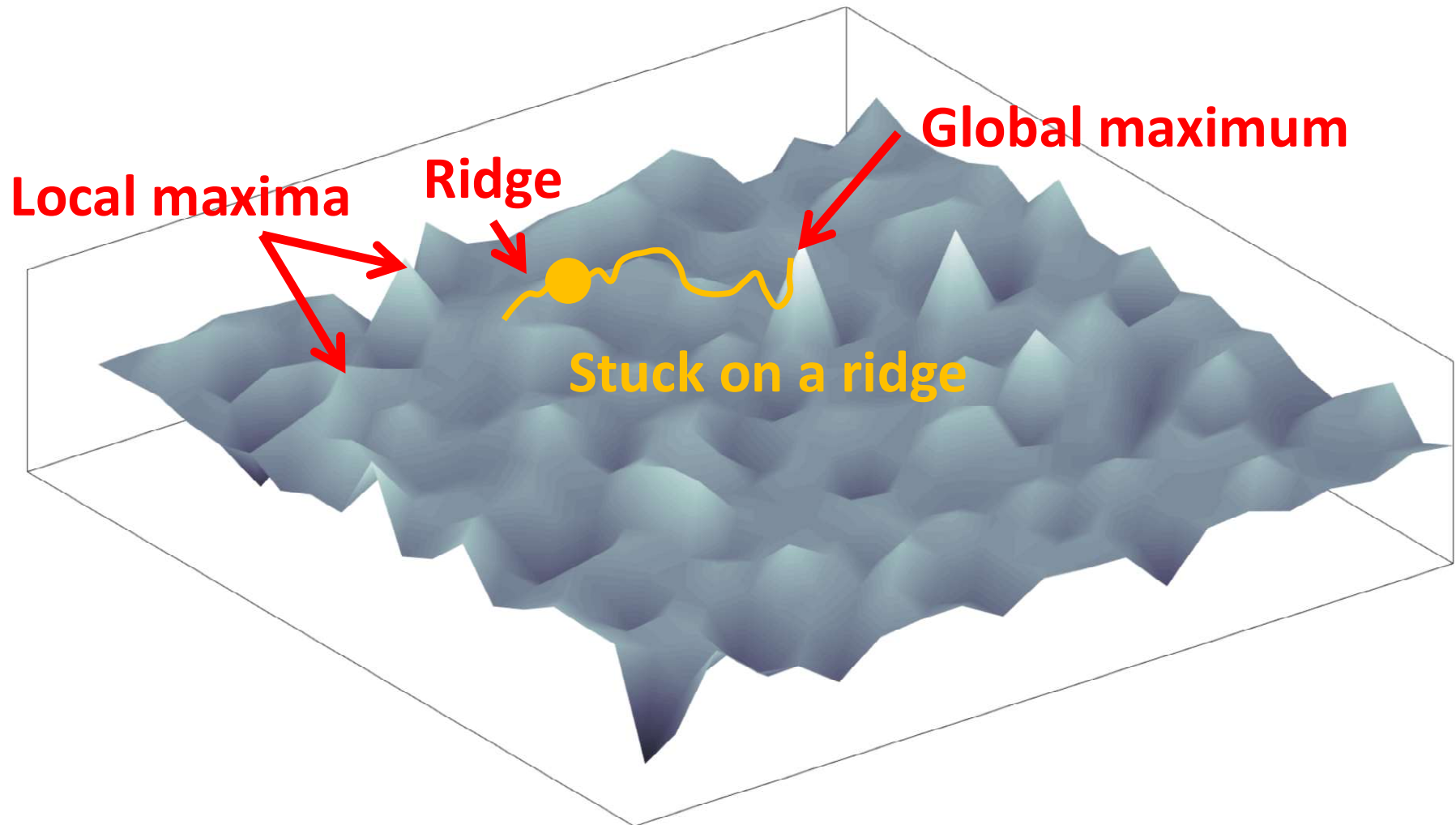
Hill Climbing Problems: Local Maxima



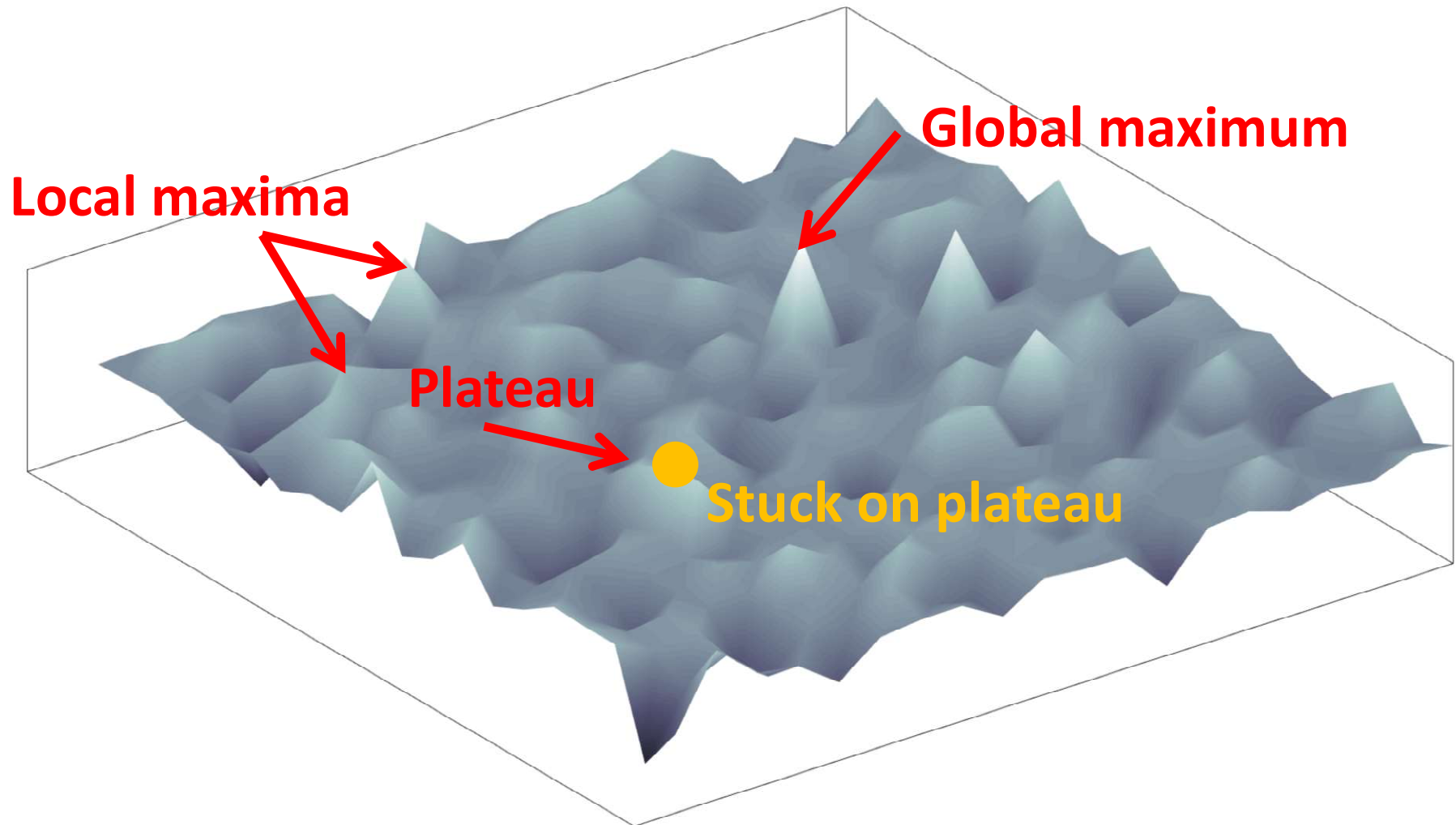
Hill Climbing Problems: Local Maxima



Hill Climbing Problems: Ridges



Hill Climbing Problems: Plateaus



Greedy Best First Search

- Also a rather primitive informed search approach
 - a naive greedy algorithm
 - evaluation function: **heuristics $h(n)$**
 - tries to not “move farther away” from the goal
 - does not care about the total path cost
- Practicalities:
 - it keep track of search history:
 - tracks visited states / nodes
 - tracks frontier states / nodes

Greedy Best First: Evaluation Function

Calculate / obtain:

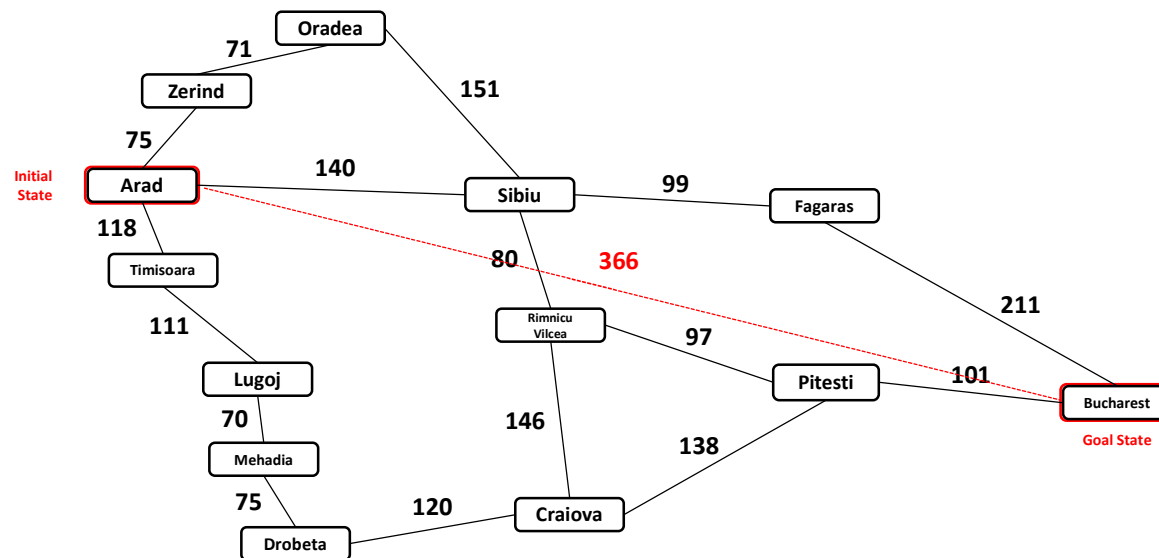
$$f(n) = h(\text{State}_n)$$

**A state n with minimum (or maximum) $f(n)$
should be chosen for expansion**

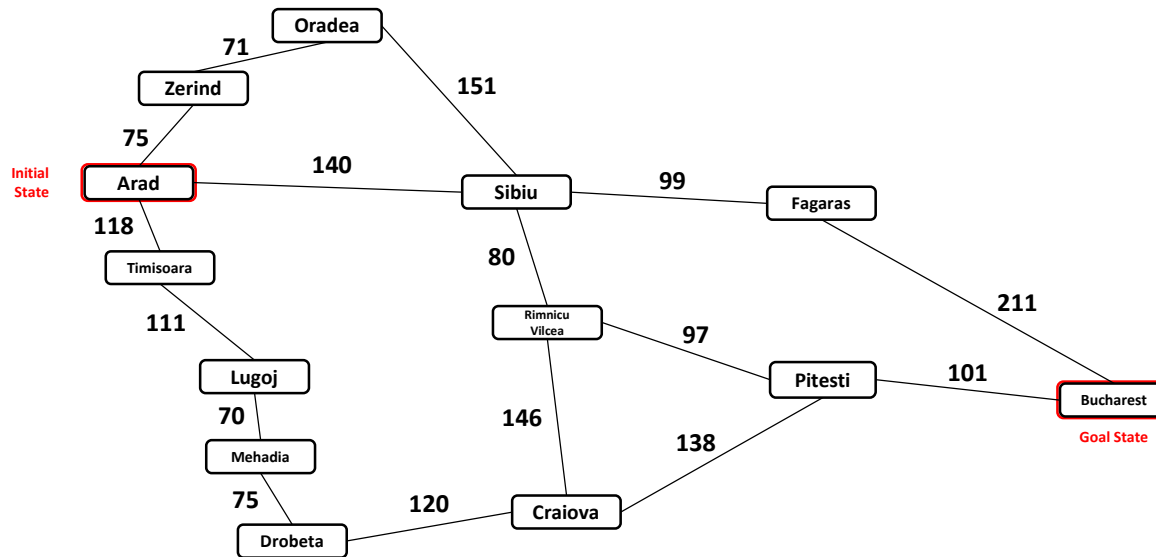
Dracula's Roadtrip: Heuristics $h(n)$

For this particular problem the heuristic function $h(n)$ is defined by a **straight-line (Euclidean) distance** between two states (cities).

“As the crows flies” in other words.



Dracula's Roadtrip: Greedy Best First



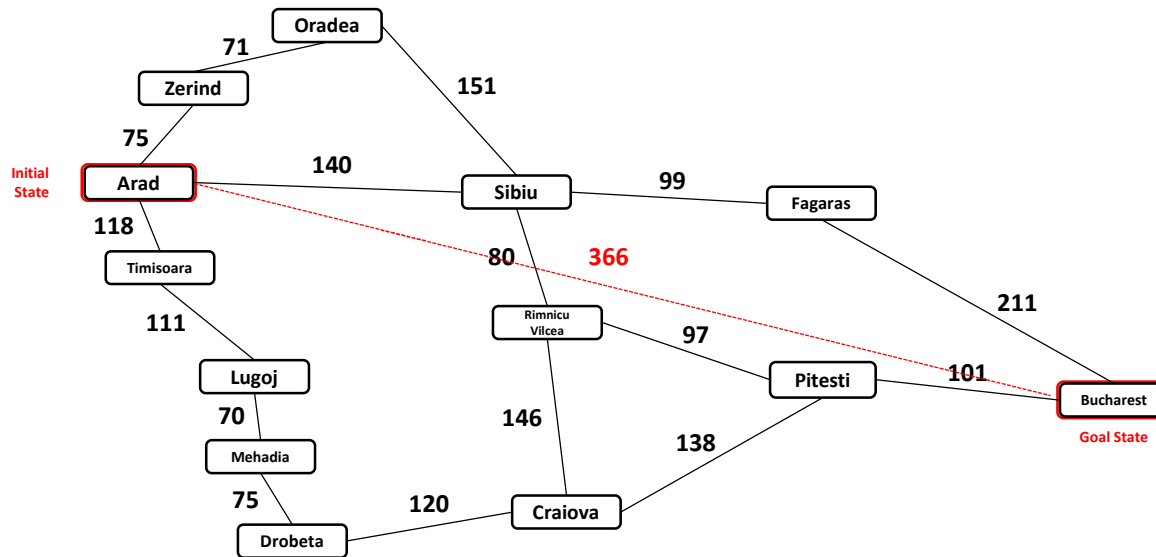
Arad

$f(\text{Arad}) = h(\text{Arad})$

Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Dracula's Roadtrip: Greedy Best First

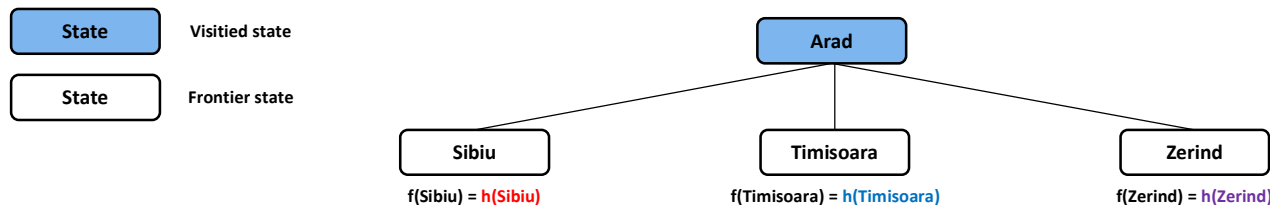
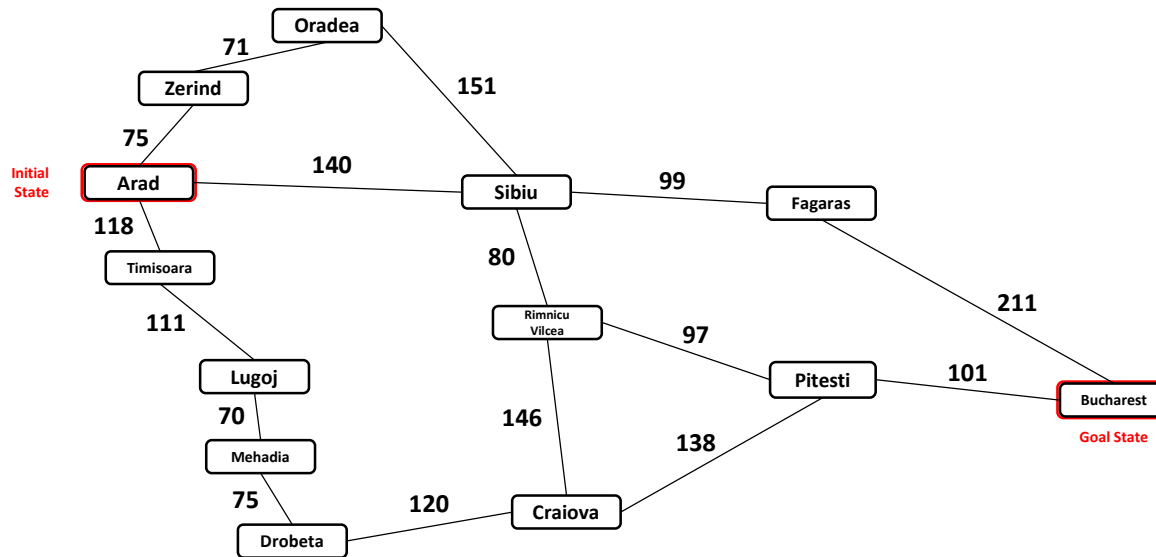


Arad
 $f(\text{Arad}) = 366$

Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

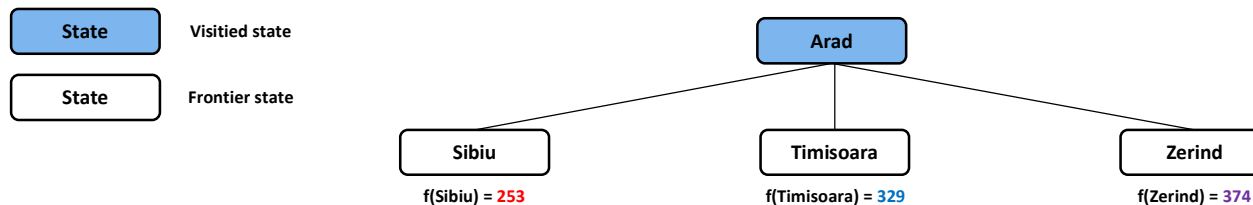
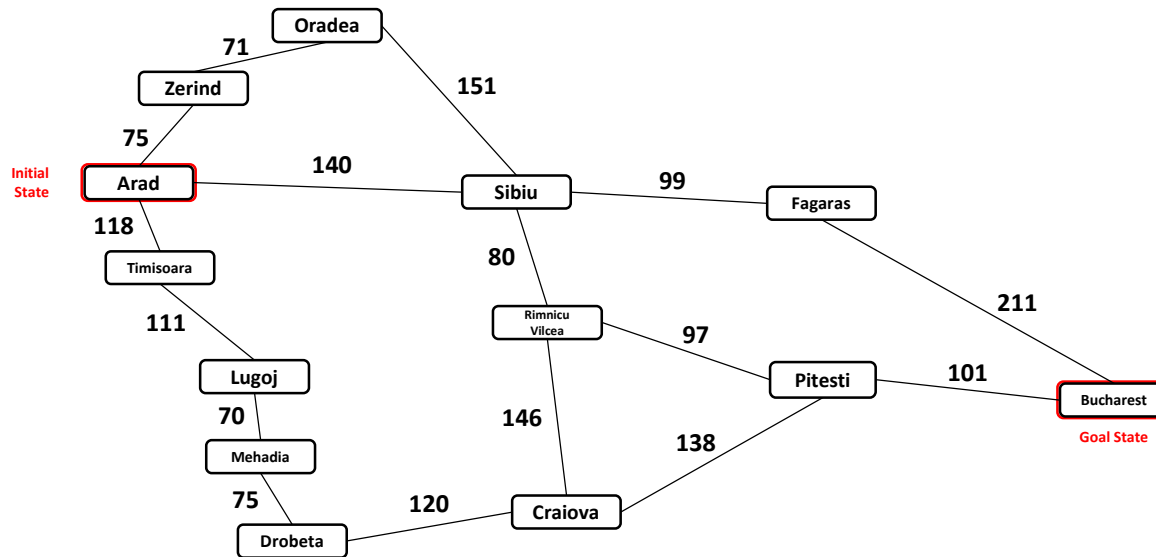
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

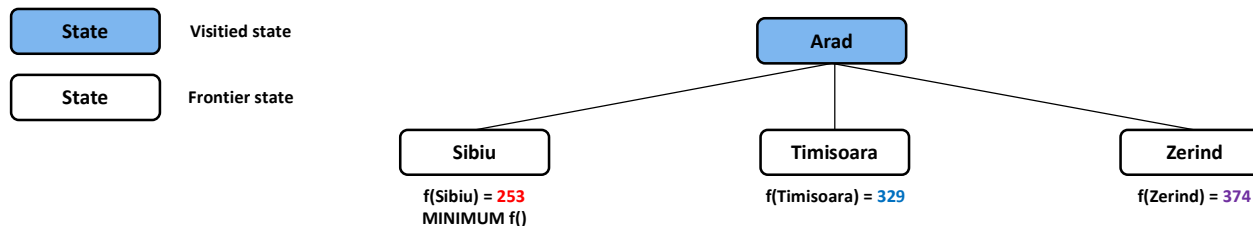
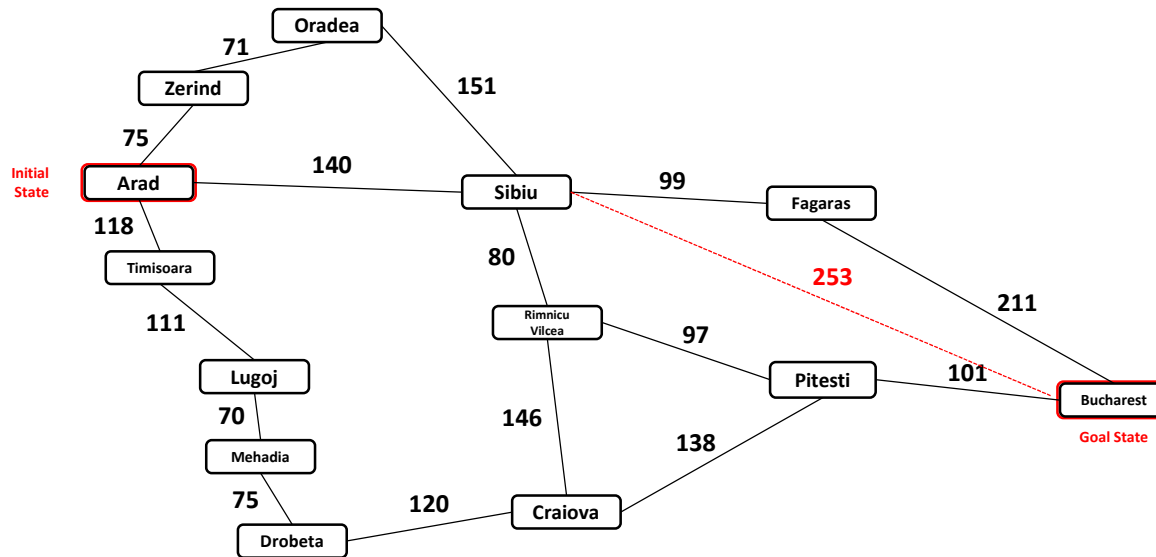
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

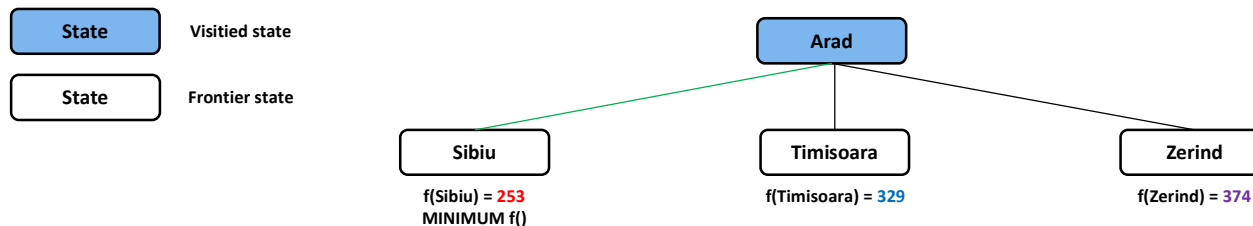
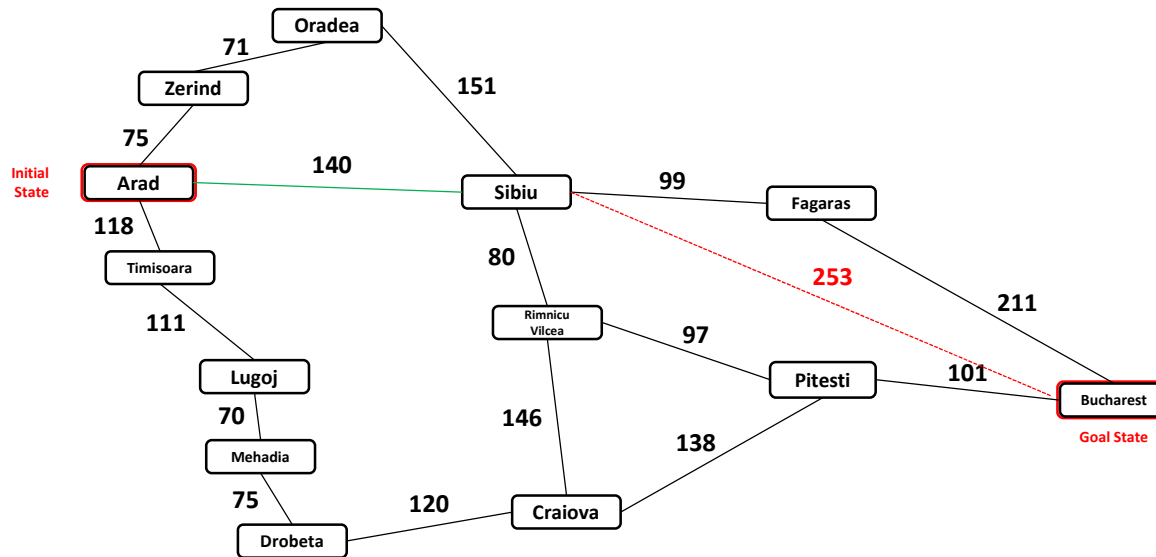
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu	
Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

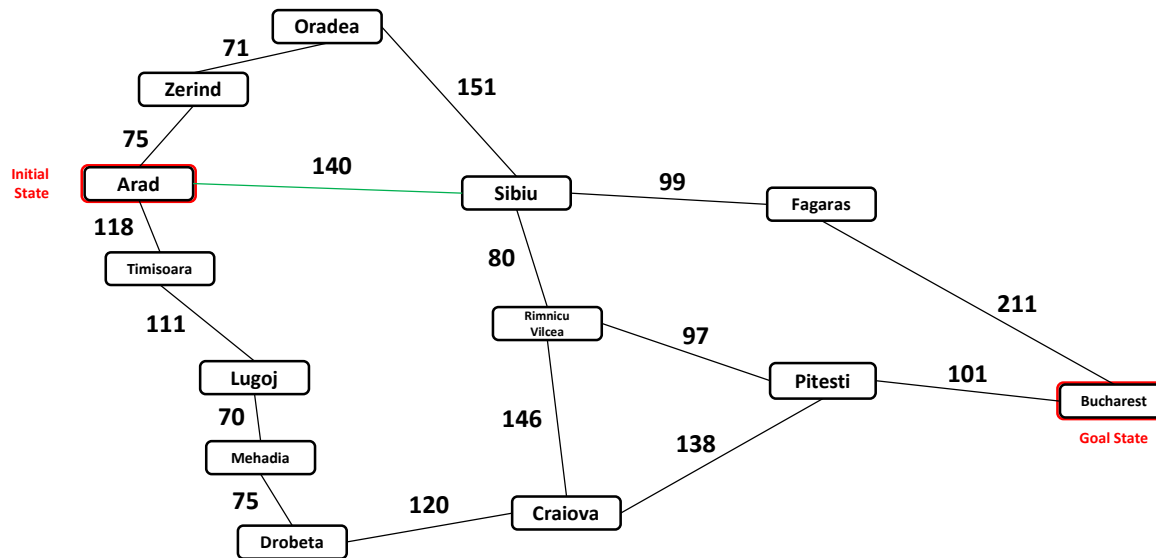
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

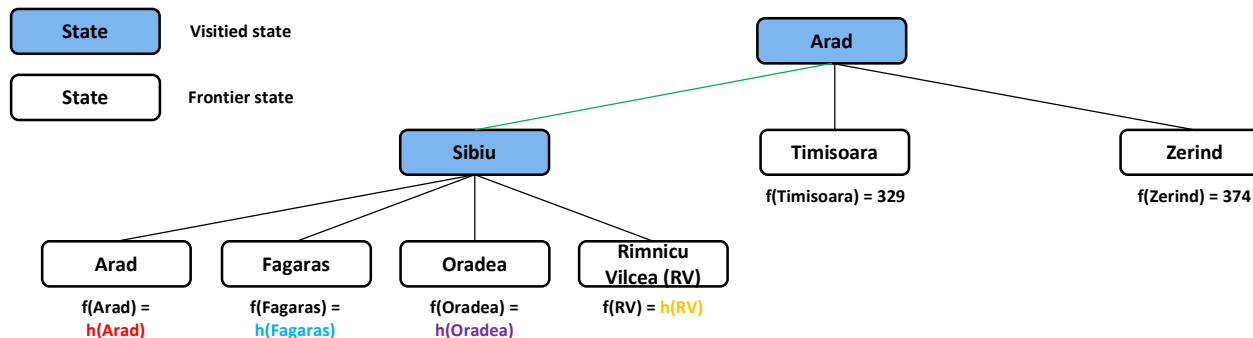
Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu	
Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Dracula's Roadtrip: Greedy Best First

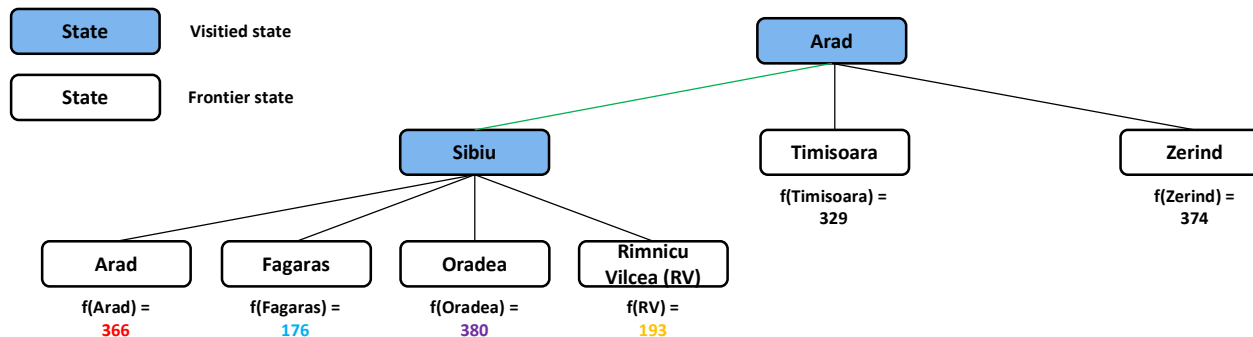
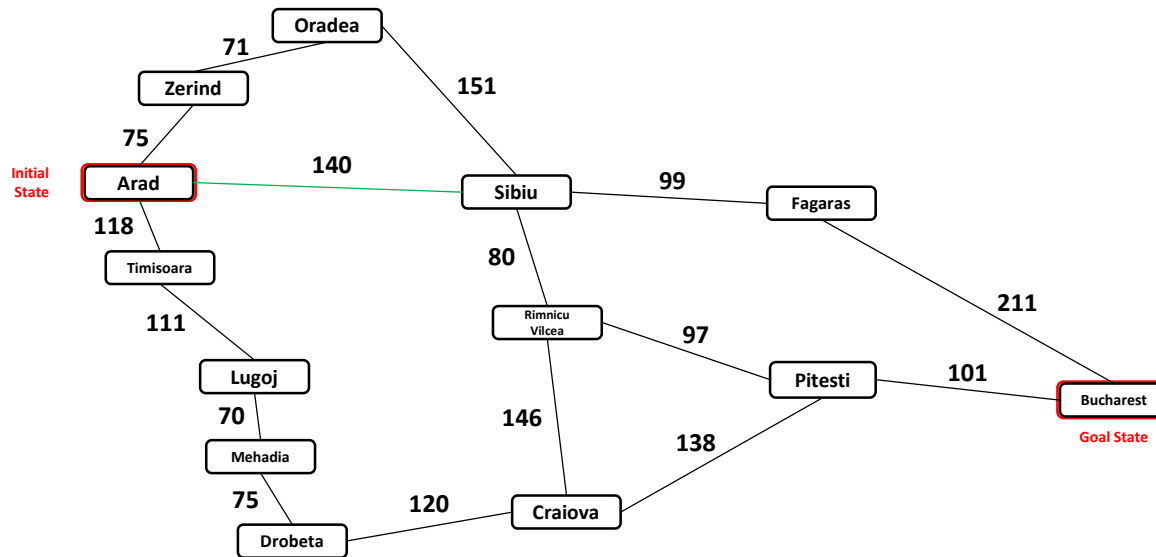


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



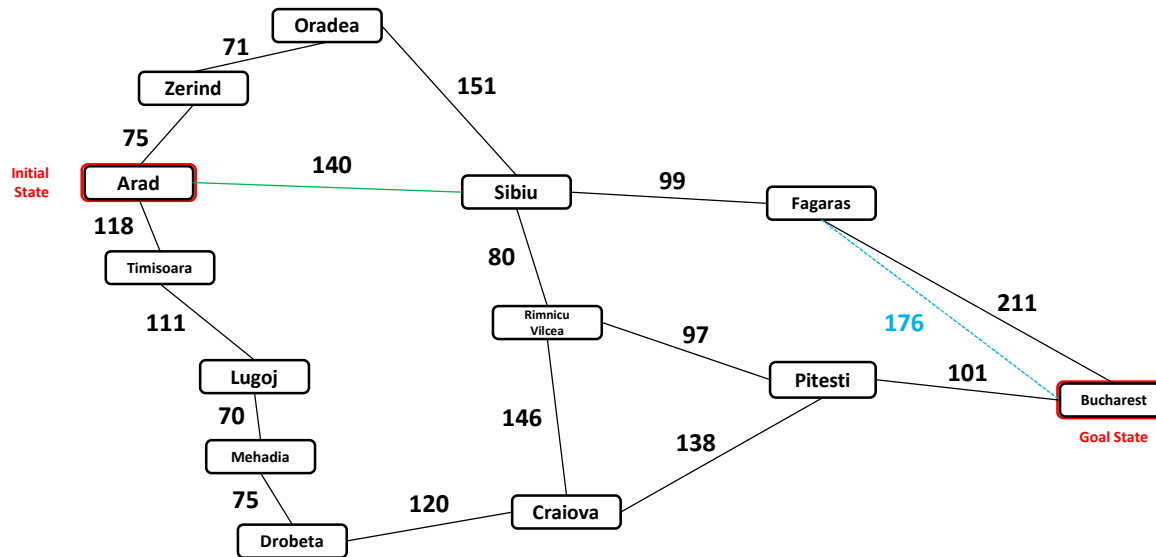
Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

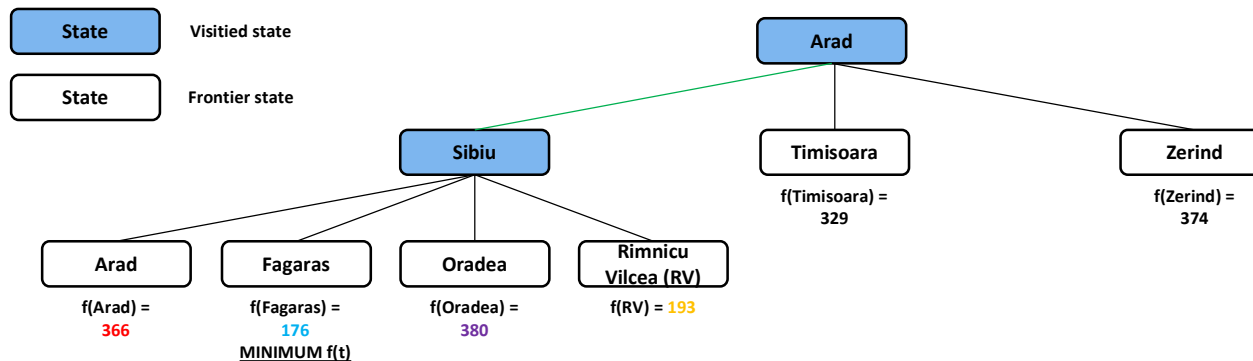
Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Dracula's Roadtrip: Greedy Best First

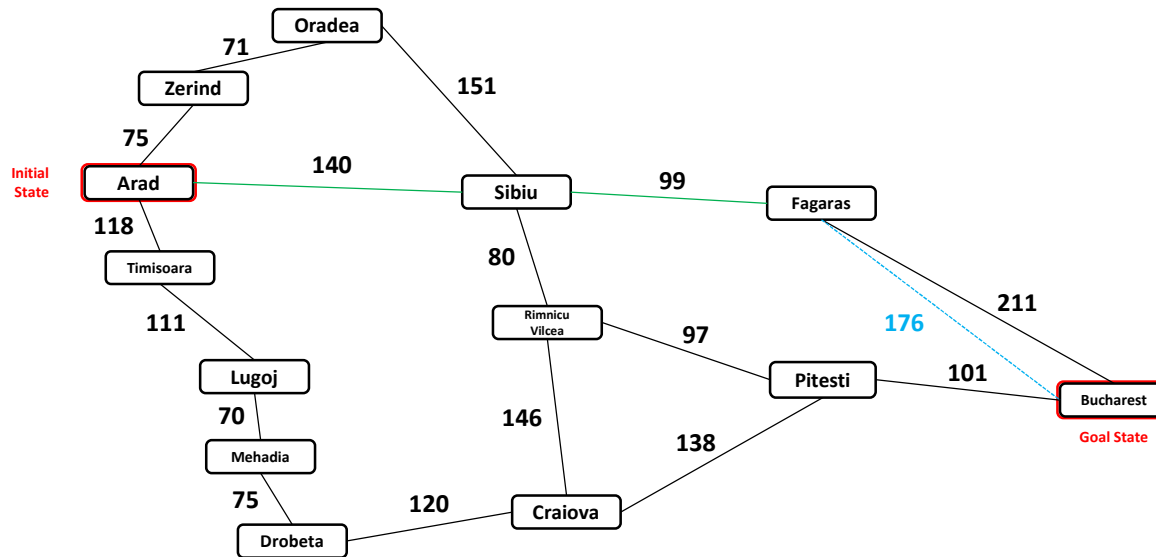


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

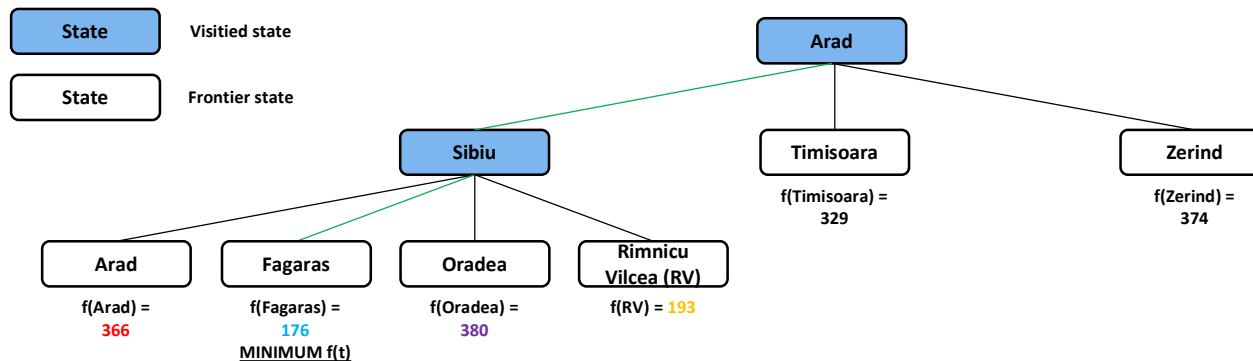


Dracula's Roadtrip: Greedy Best First

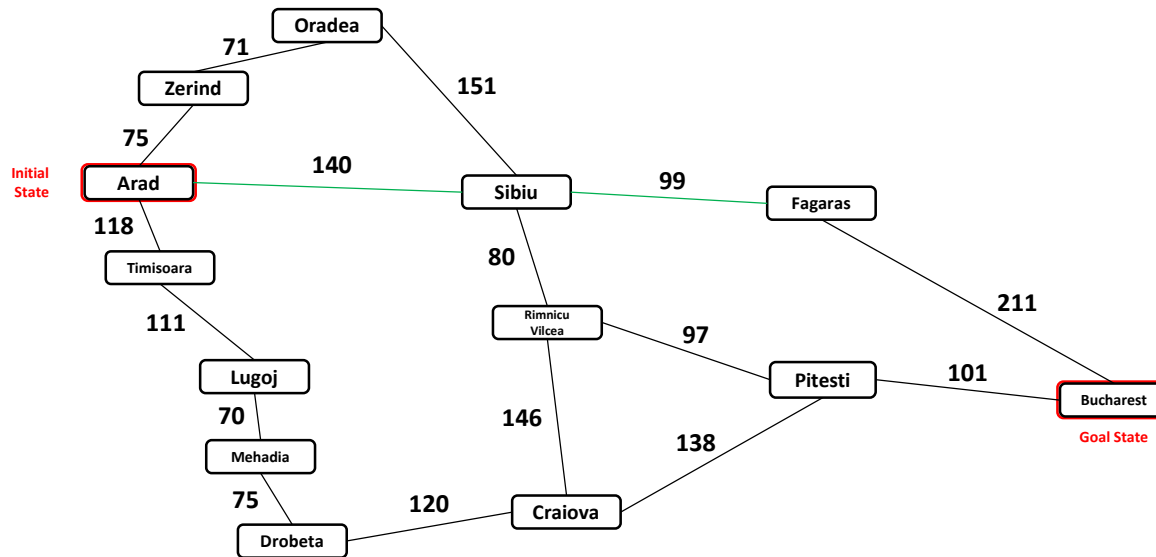


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

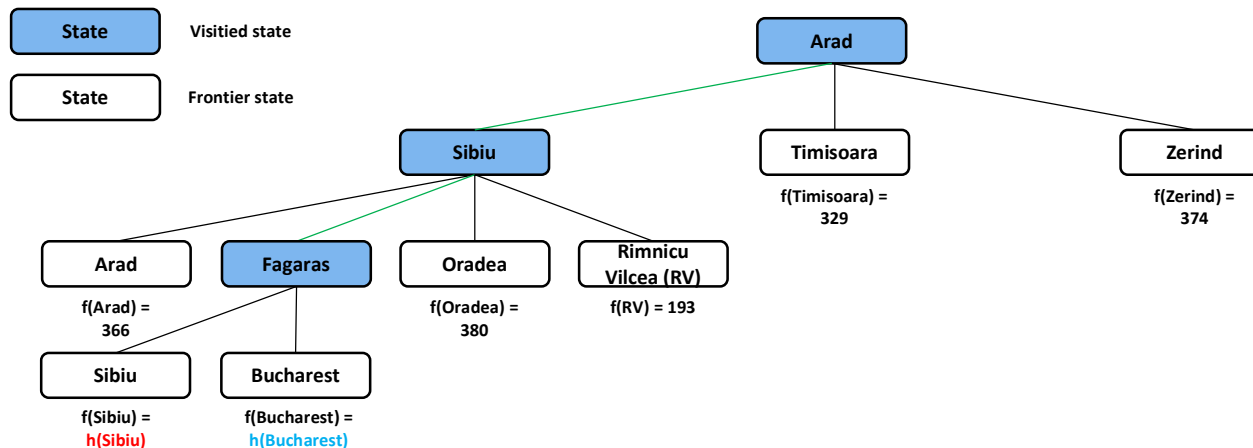


Dracula's Roadtrip: Greedy Best First

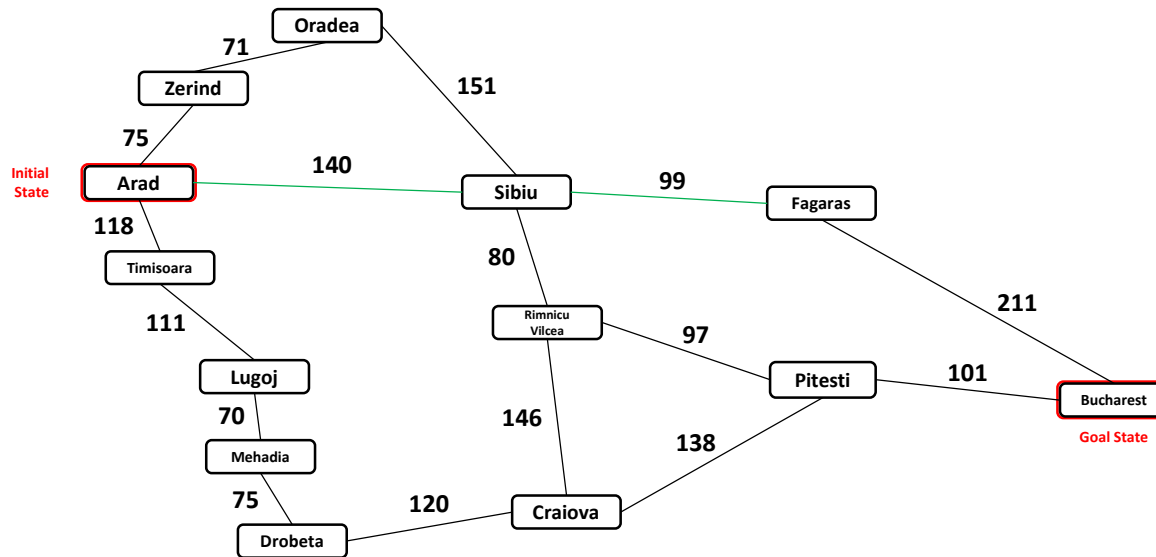


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

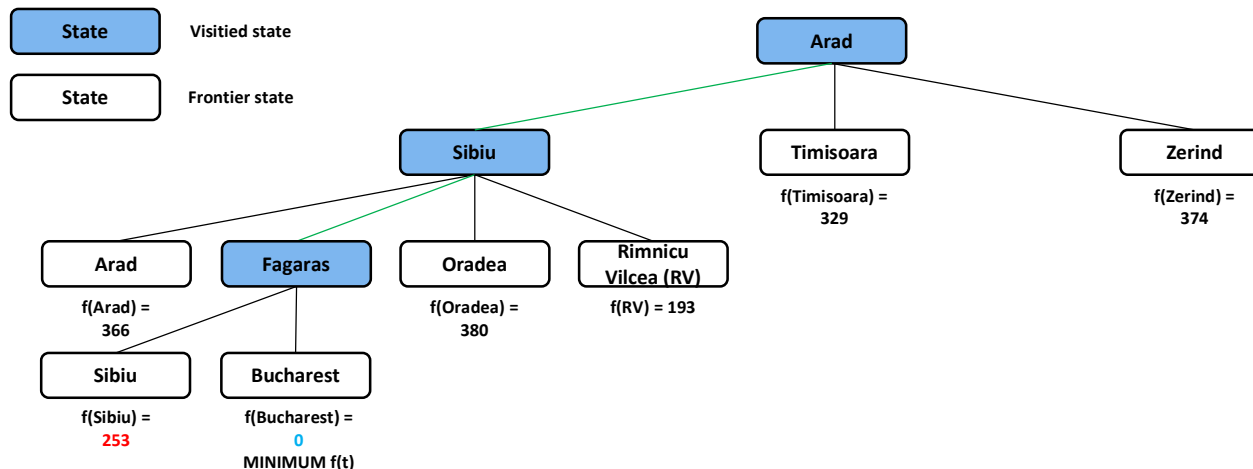


Dracula's Roadtrip: Greedy Best First

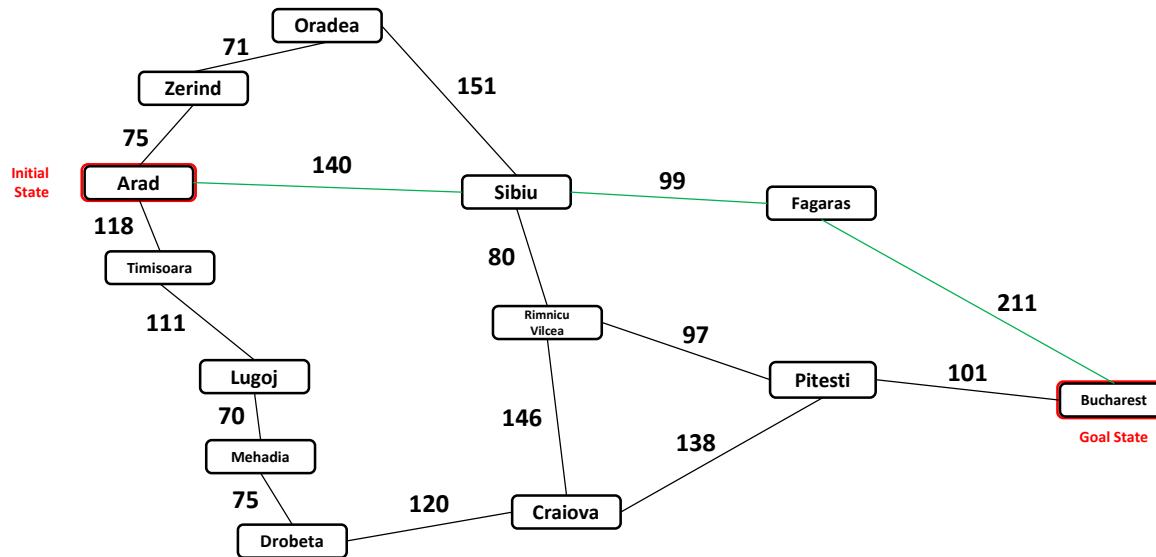


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

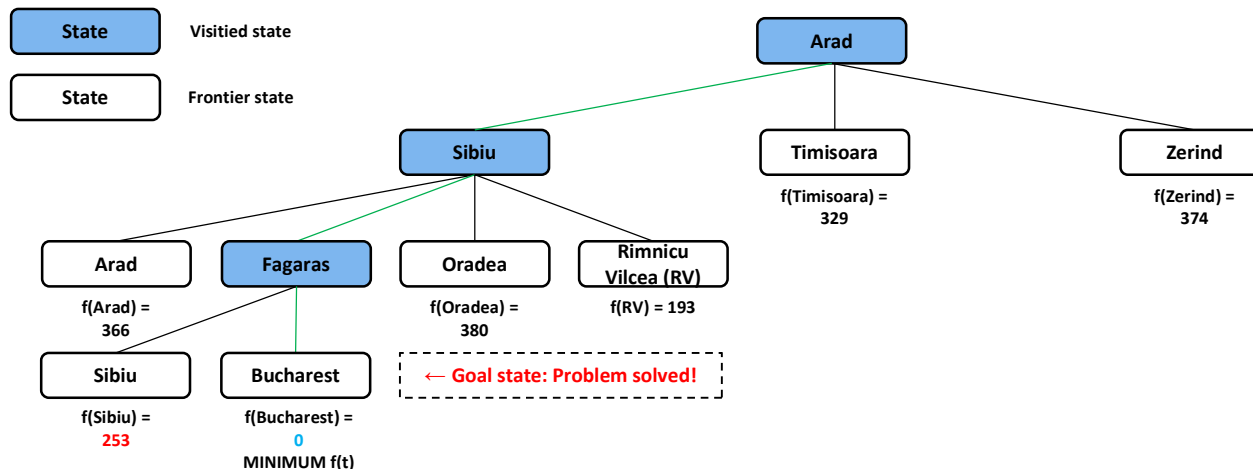


Dracula's Roadtrip: Greedy Best First



Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



A* Algorithm: Evaluation Function

Calculate / obtain:

$$f(n) = g(\text{State}_n) + h(\text{State}_n)$$

where:

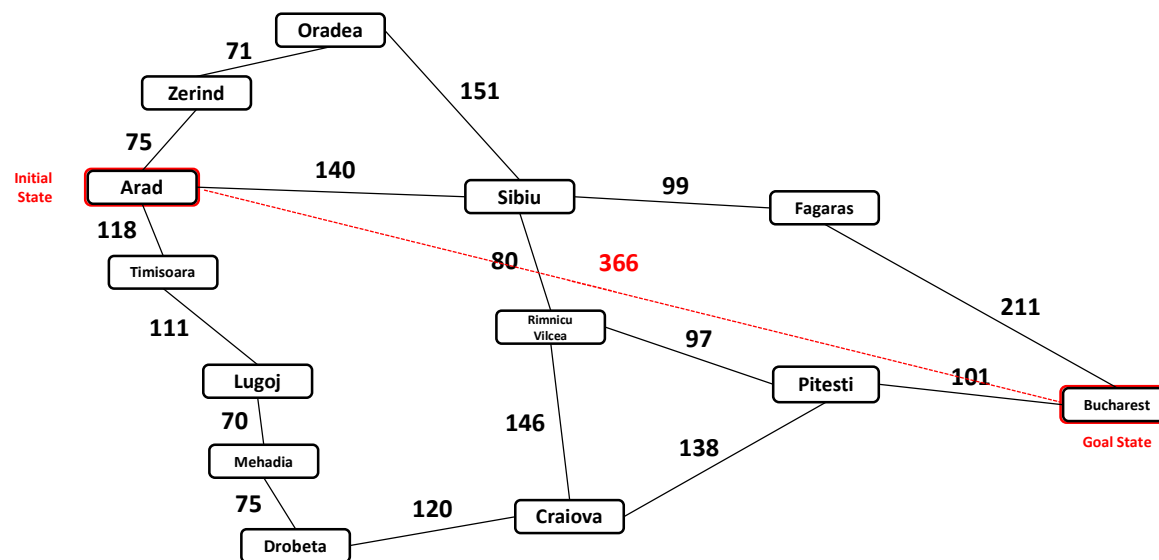
- $g(n)$ - initial node to node n path cost
- $h(n)$ - **estimated cost** of the best path that continues from node n to a goal node

A state n with minimum (maximum) $f(n)$
should be chosen for expansion

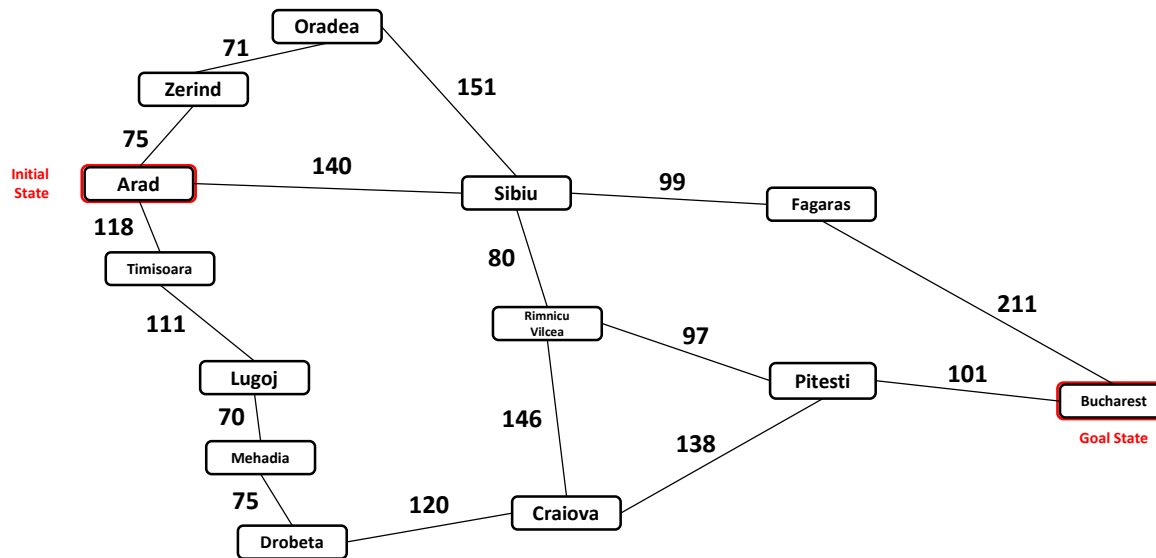
Dracula's Roadtrip: Heuristics $h(n)$

For this particular problem the heuristic function $h(n)$ is defined by a **straight-line (Euclidean) distance** between two states (cities).

“As the crows flies” in other words.



Dracula's Roadtrip: A* Search



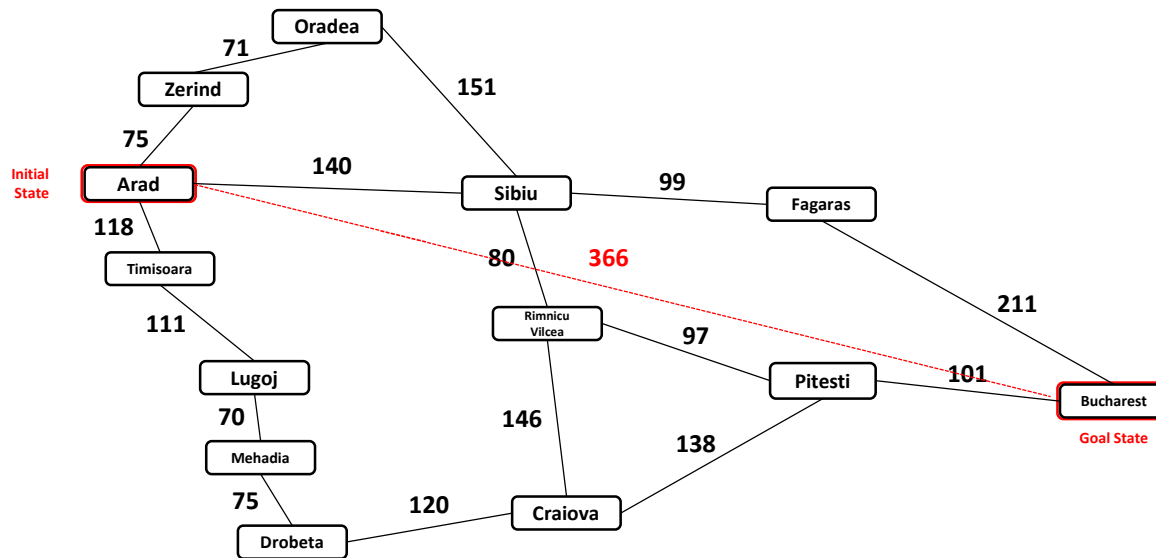
Arad

$$f(\text{Arad}) = g(\text{Arad}) + h(\text{Arad})$$

Straight-line distance to Bucharest ($h(\text{State})$):

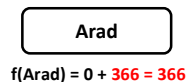
Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Dracula's Roadtrip: A* Search

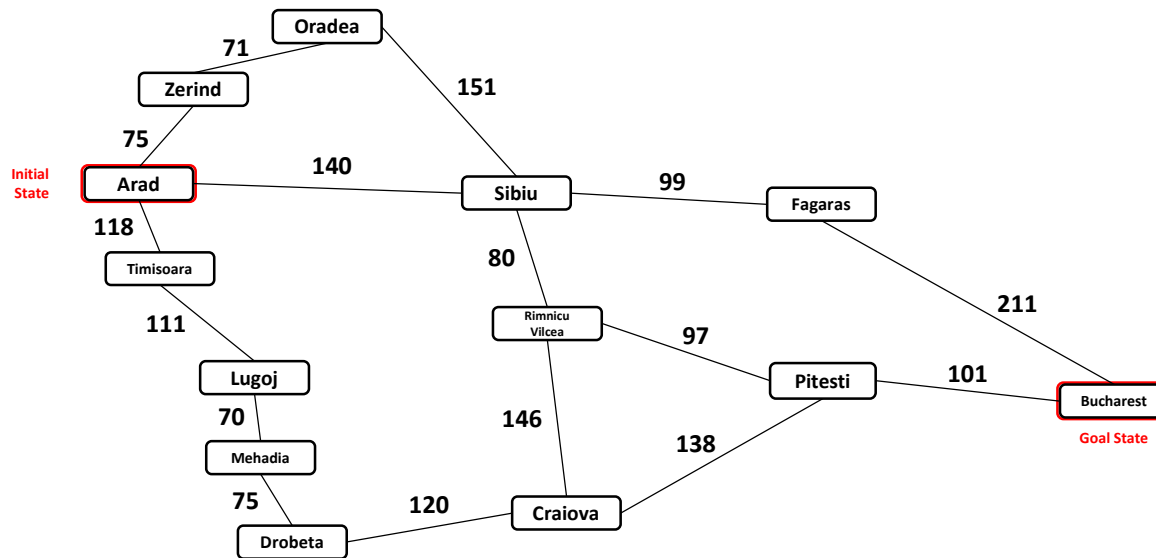


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

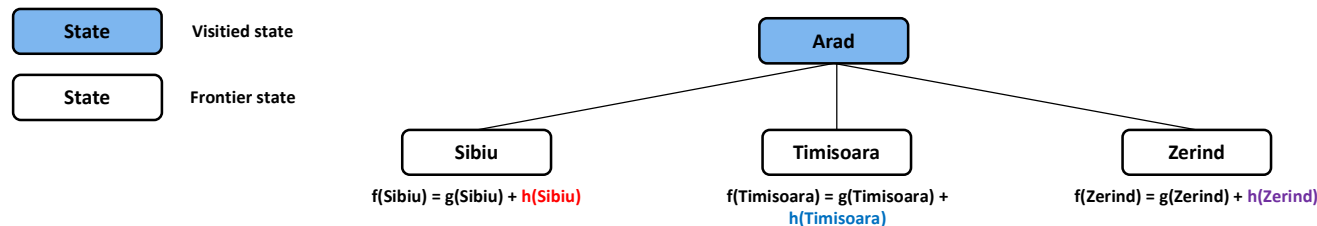


Dracula's Roadtrip: A* Search

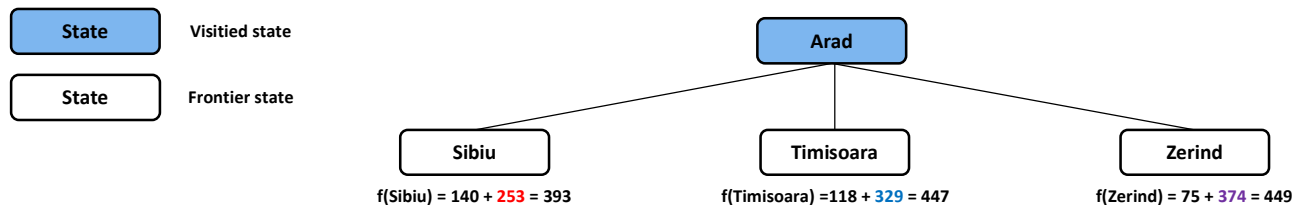
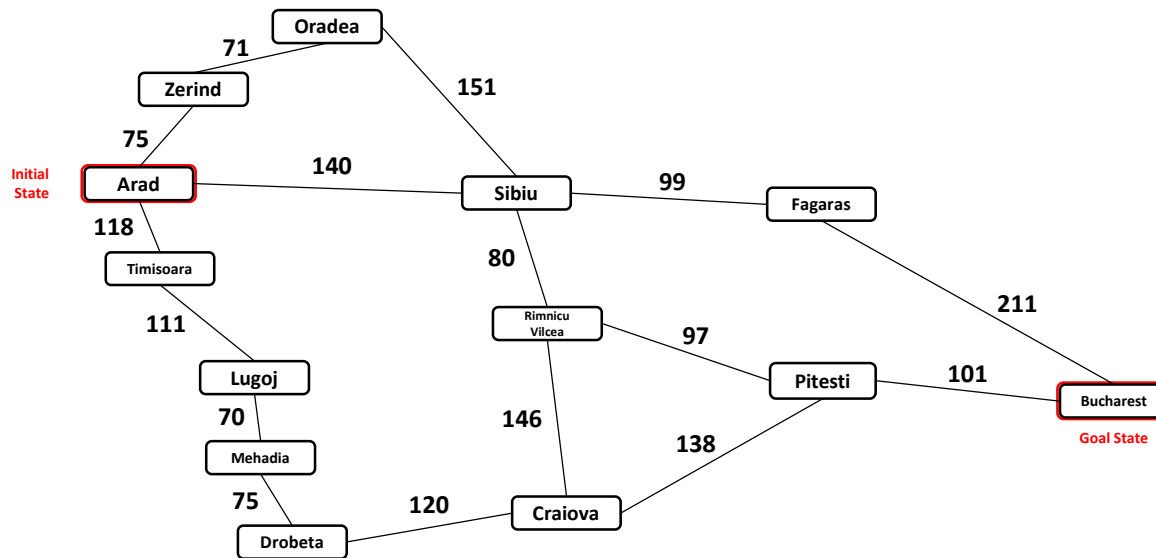


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



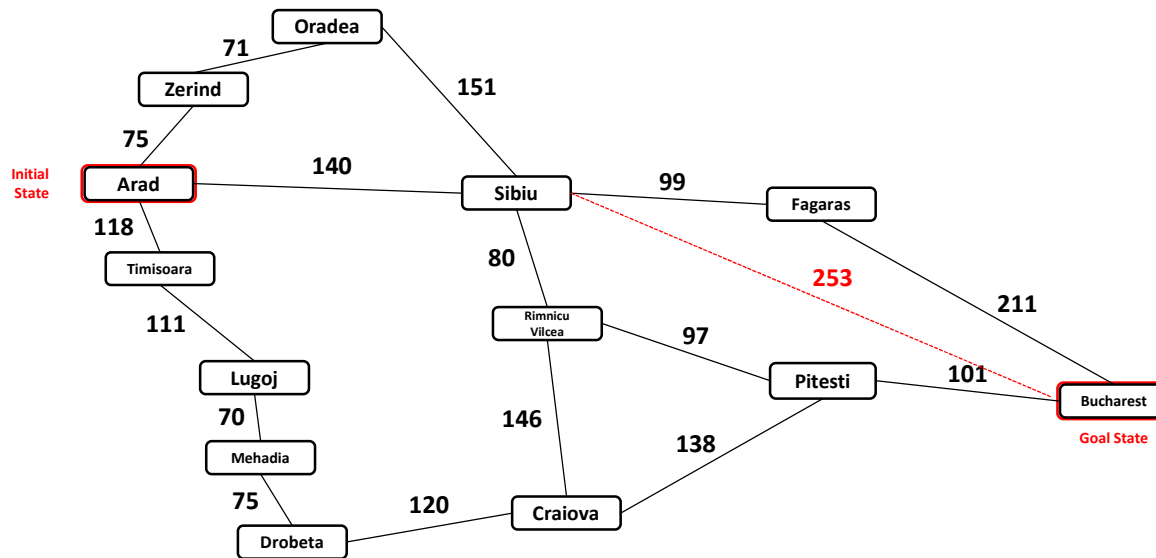
Dracula's Roadtrip: A* Search



Straight-line distance to Bucharest ($h(\text{State})$):

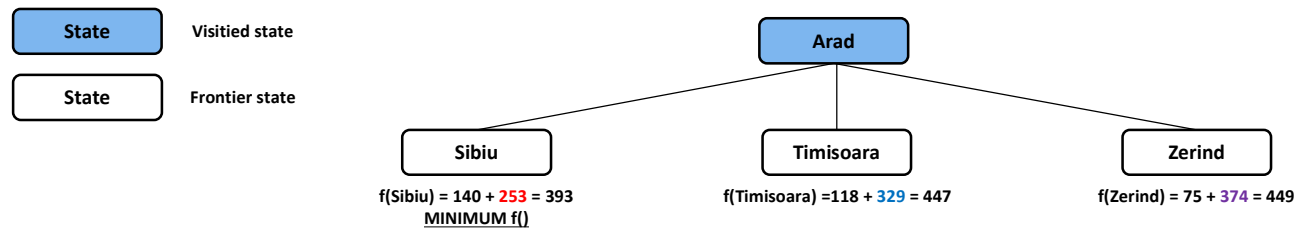
Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Dracula's Roadtrip: A* Search

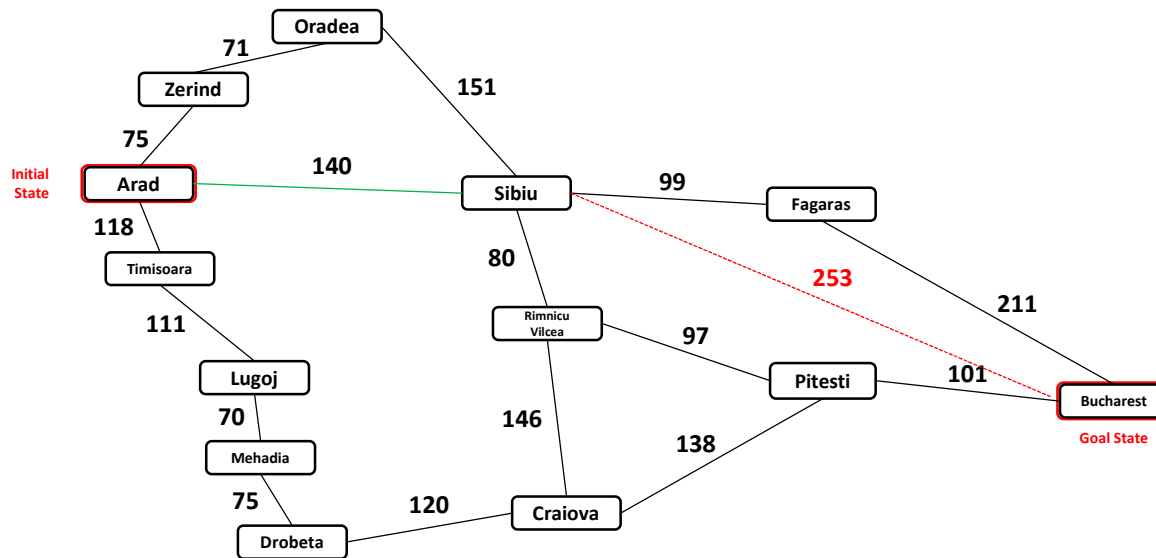


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

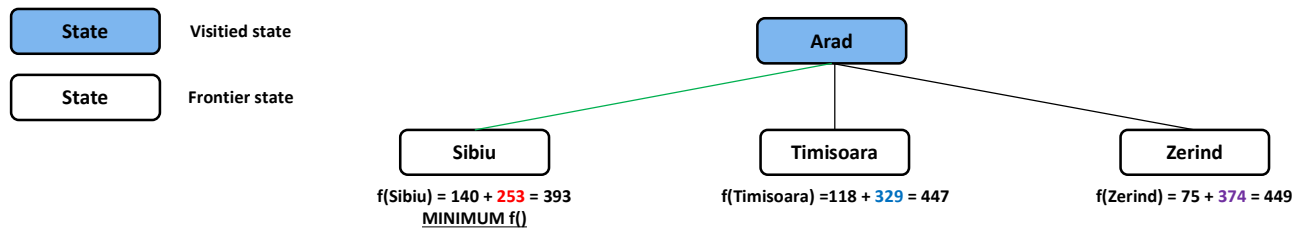


Dracula's Roadtrip: A* Search

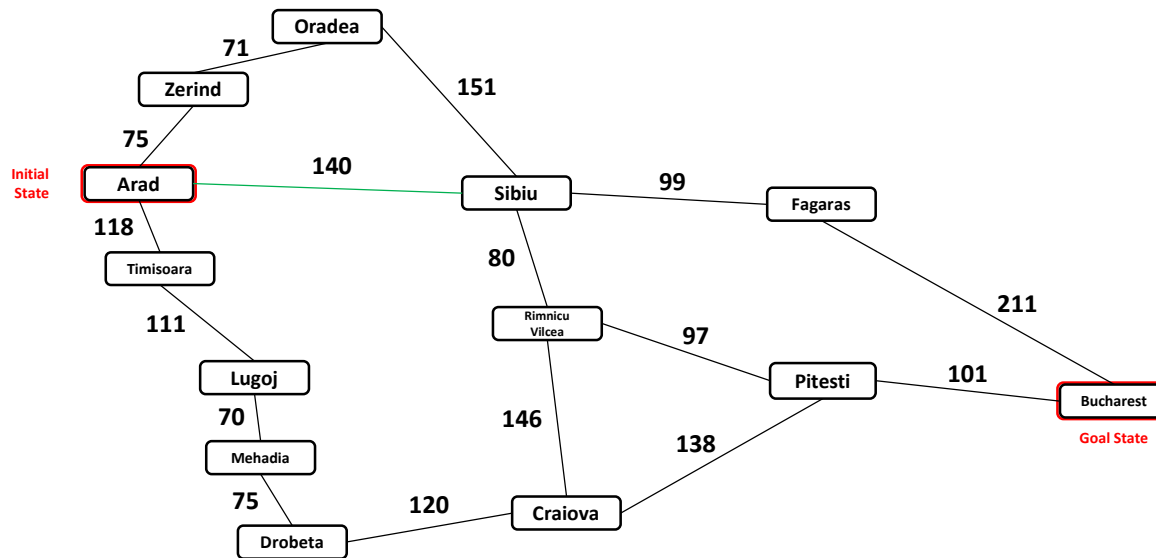


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

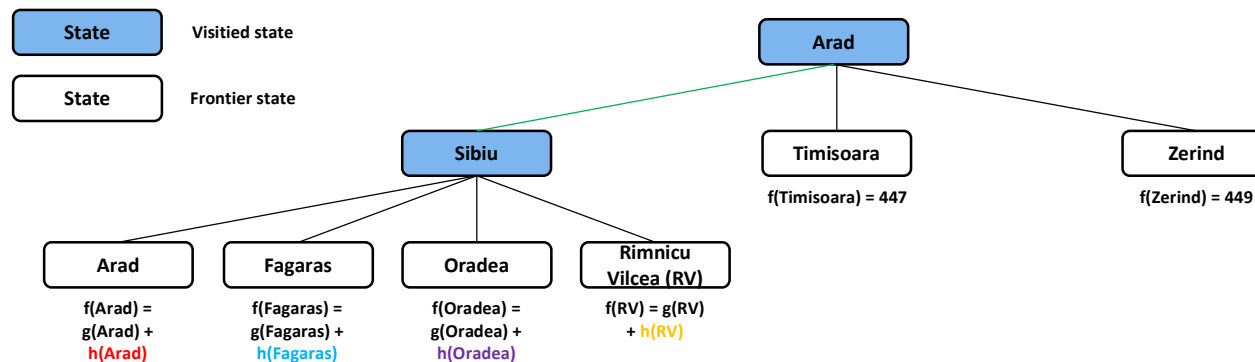


Dracula's Roadtrip: A* Search

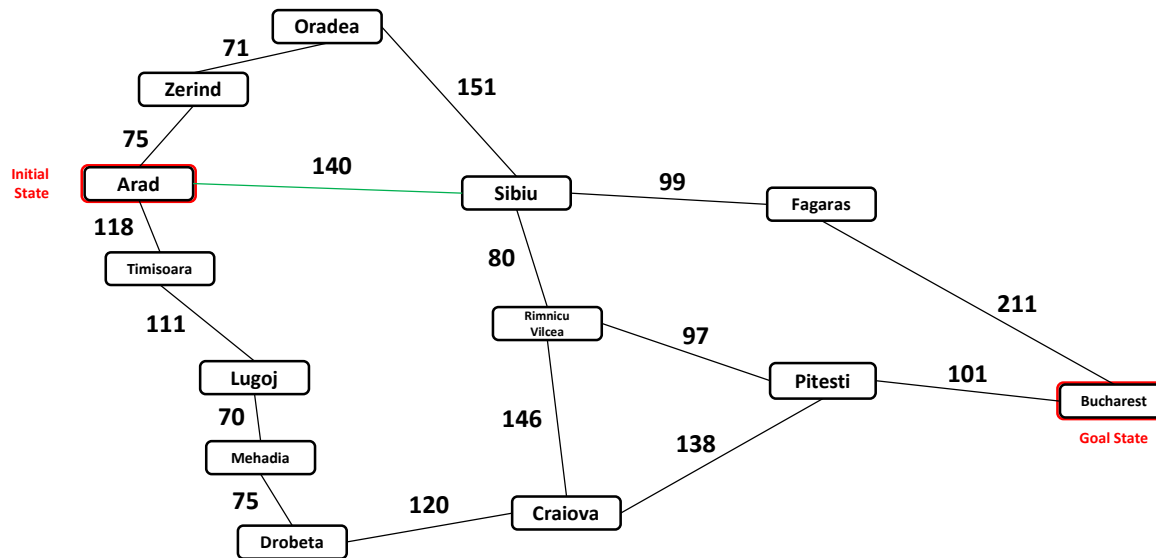


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

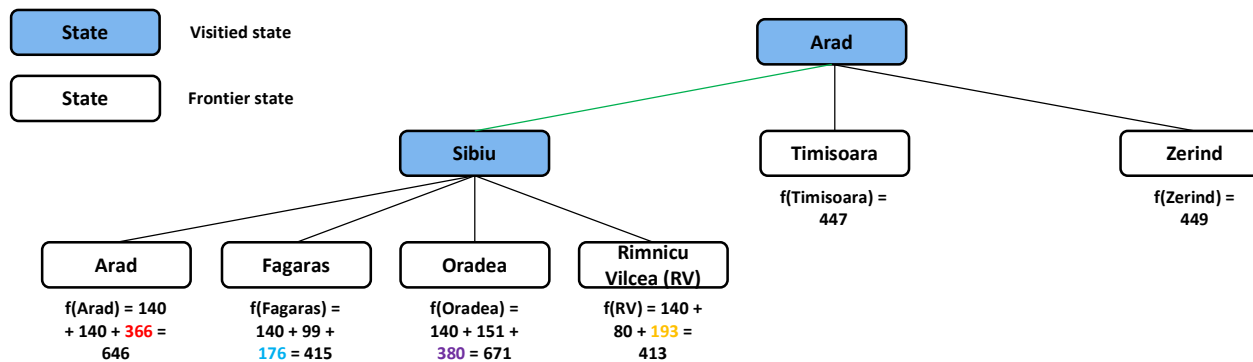


Dracula's Roadtrip: A* Search

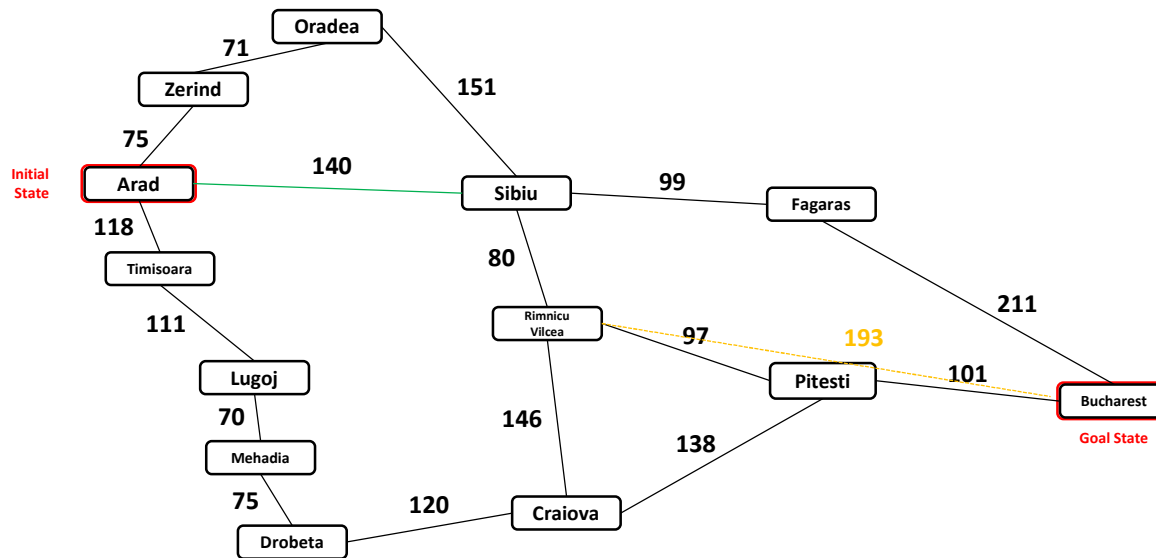


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

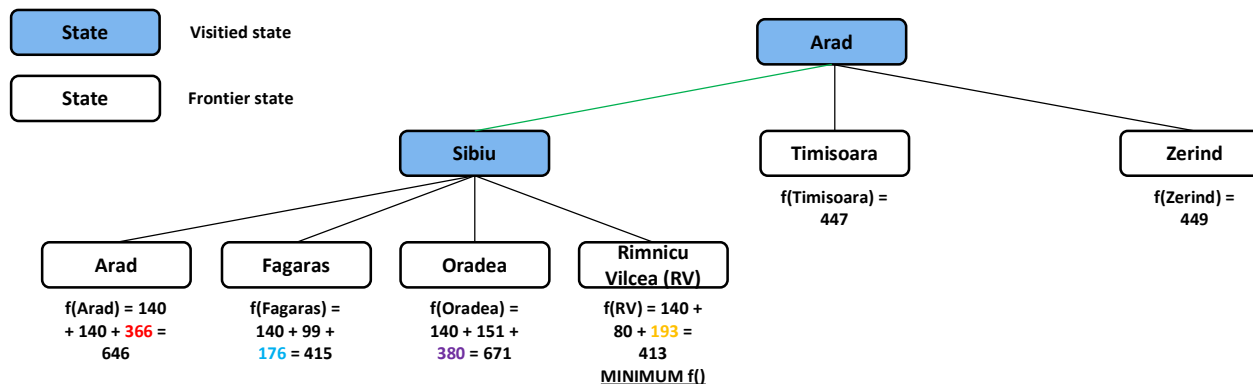


Dracula's Roadtrip: A* Search

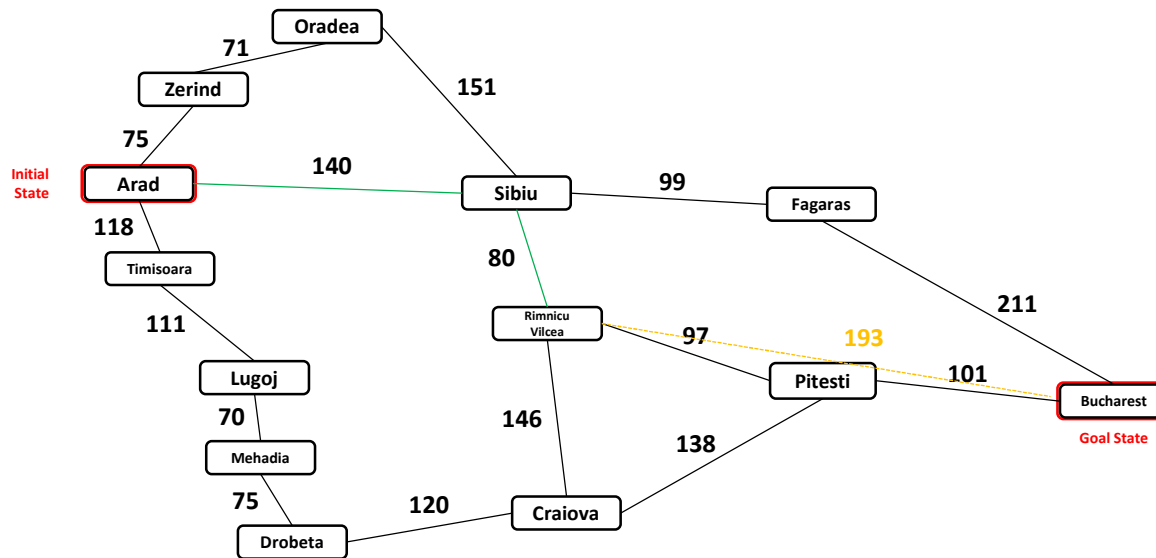


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

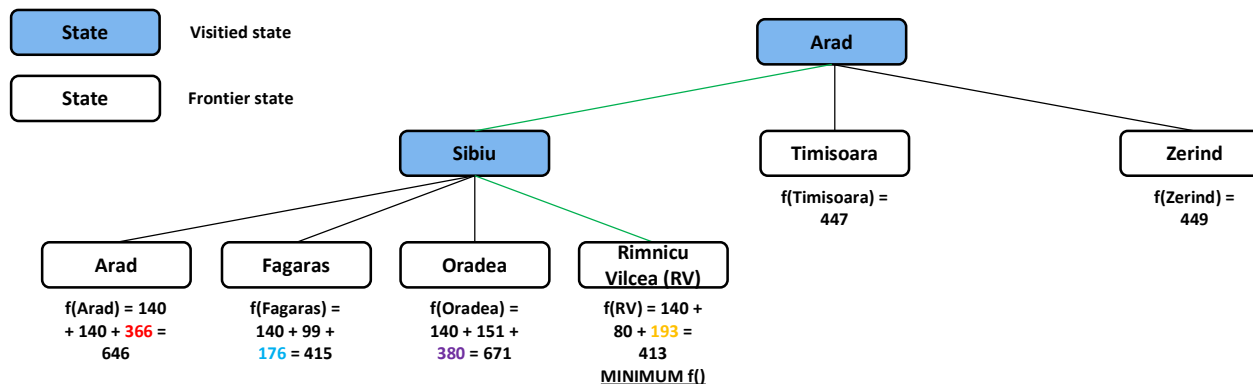


Dracula's Roadtrip: A* Search

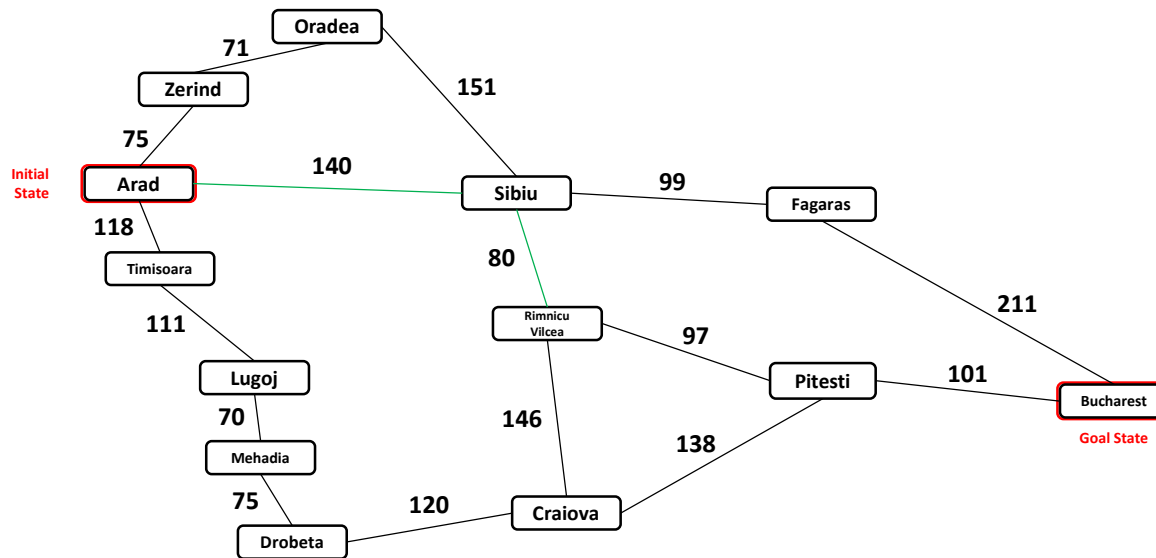


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

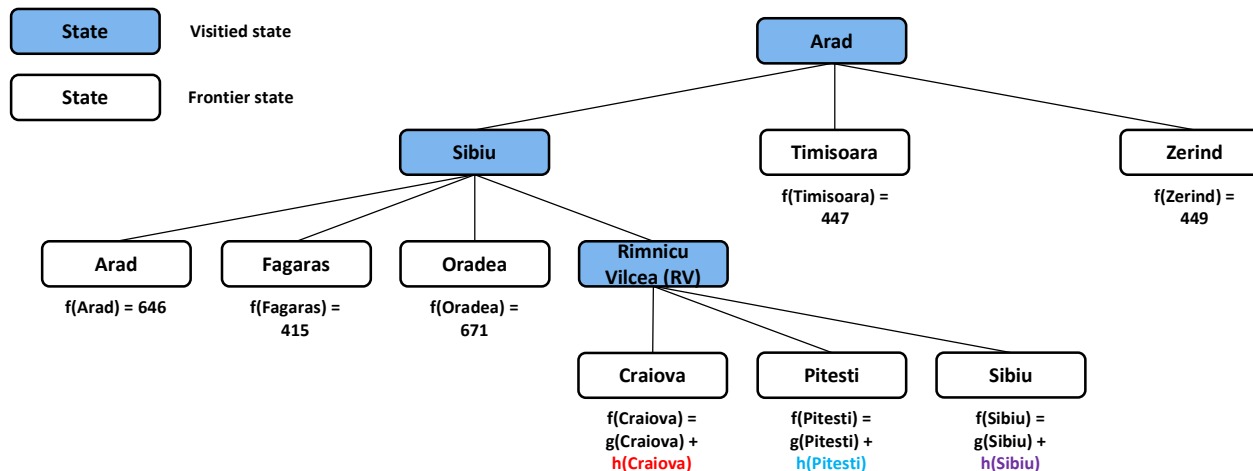


Dracula's Roadtrip: A* Search

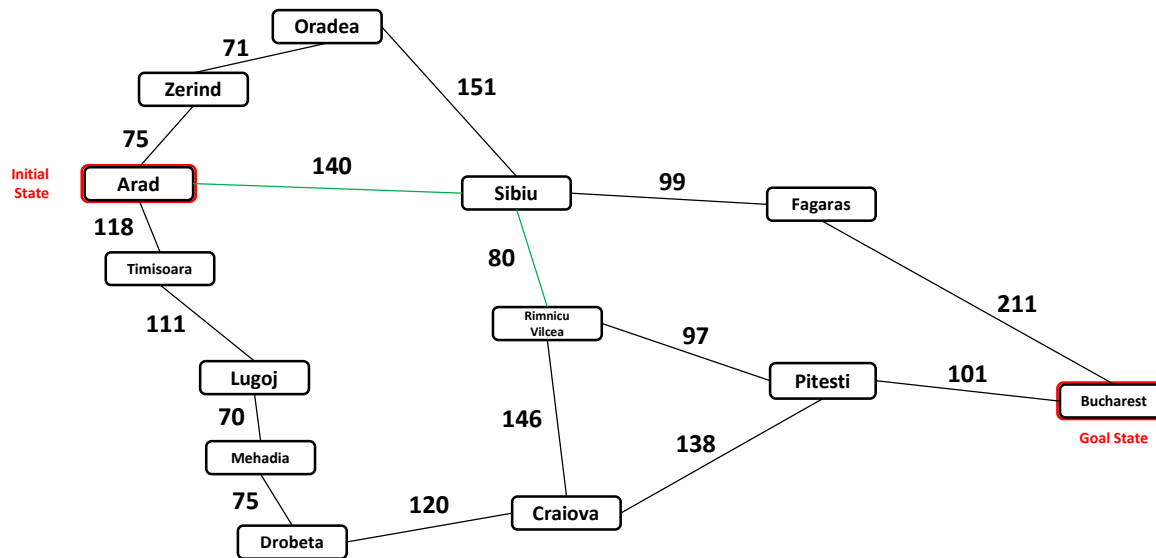


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

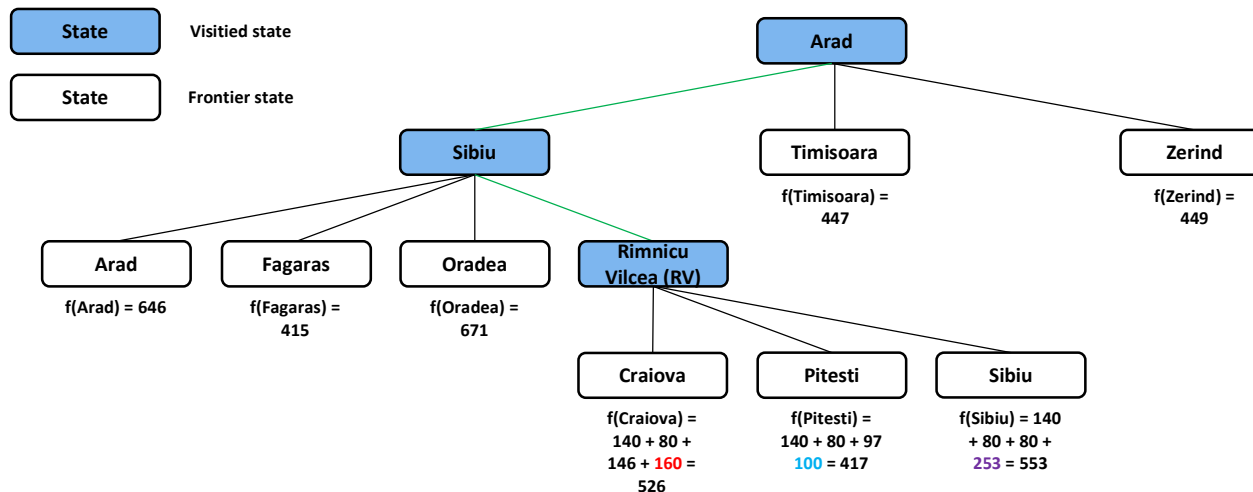


Dracula's Roadtrip: A* Search

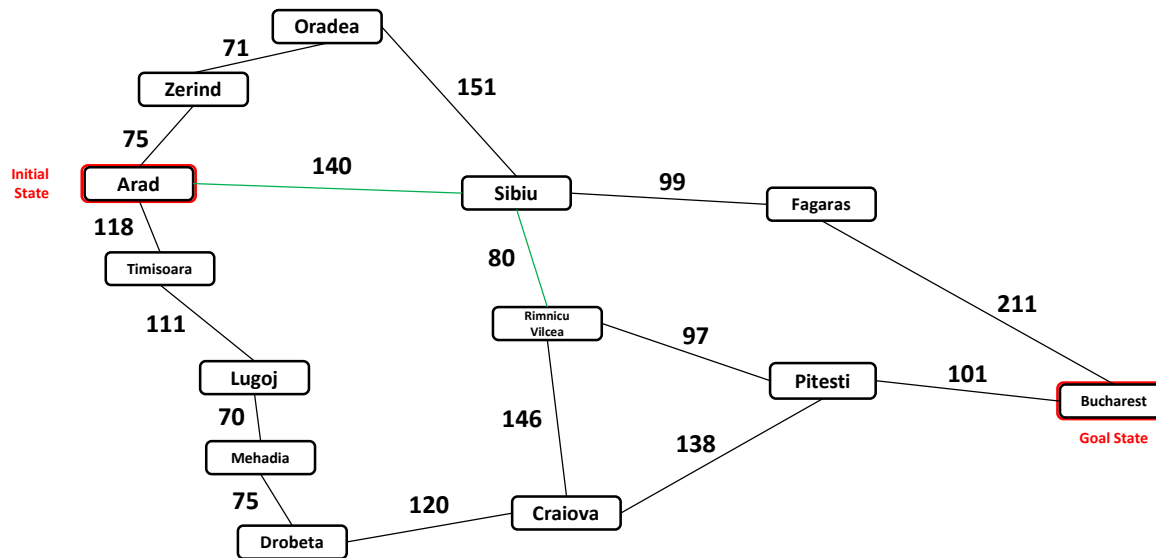


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

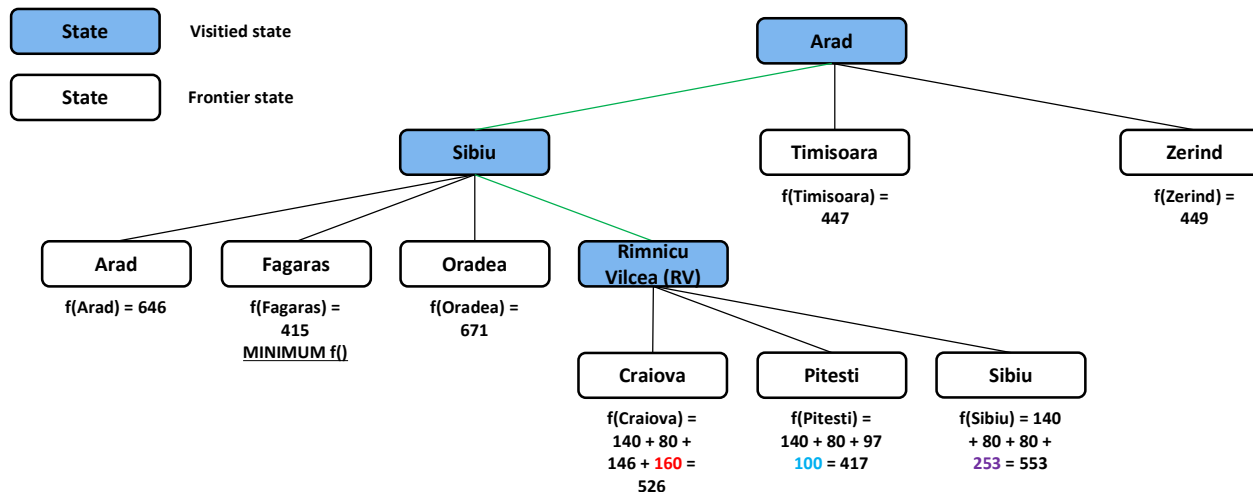


Dracula's Roadtrip: A* Search

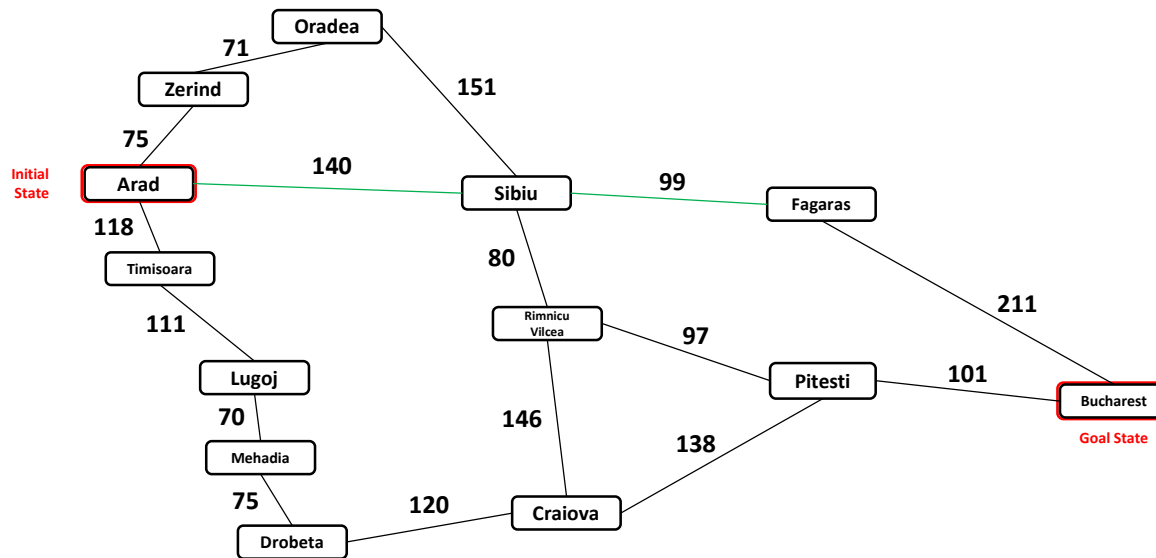


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

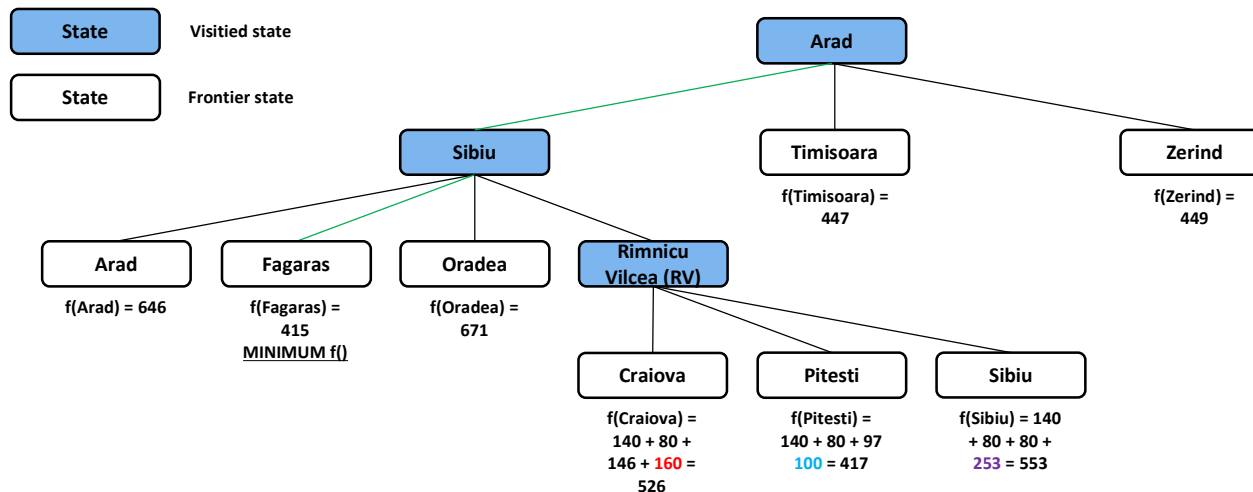


Dracula's Roadtrip: A* Search

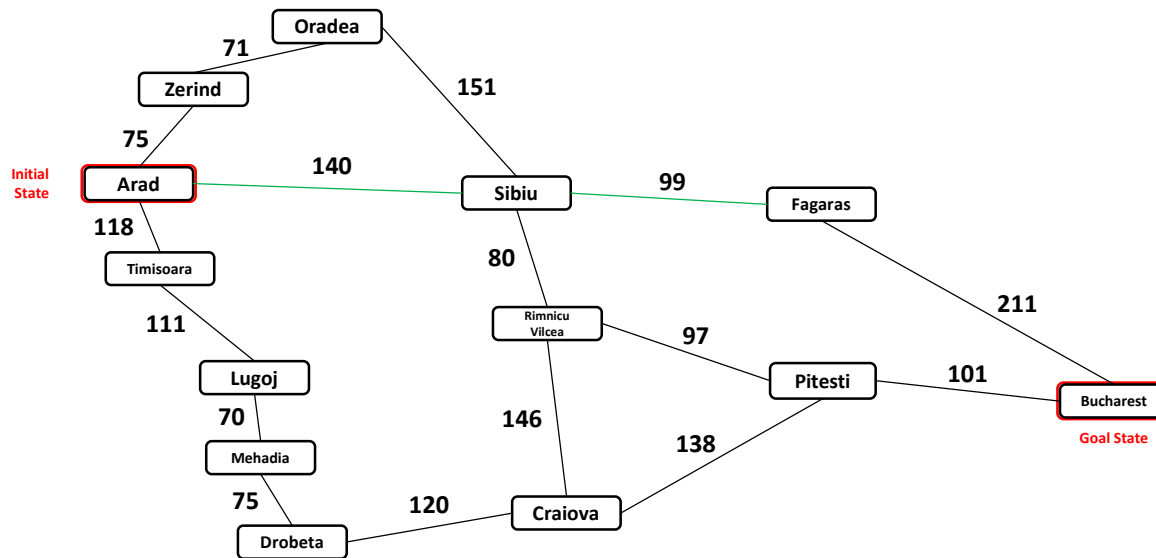


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

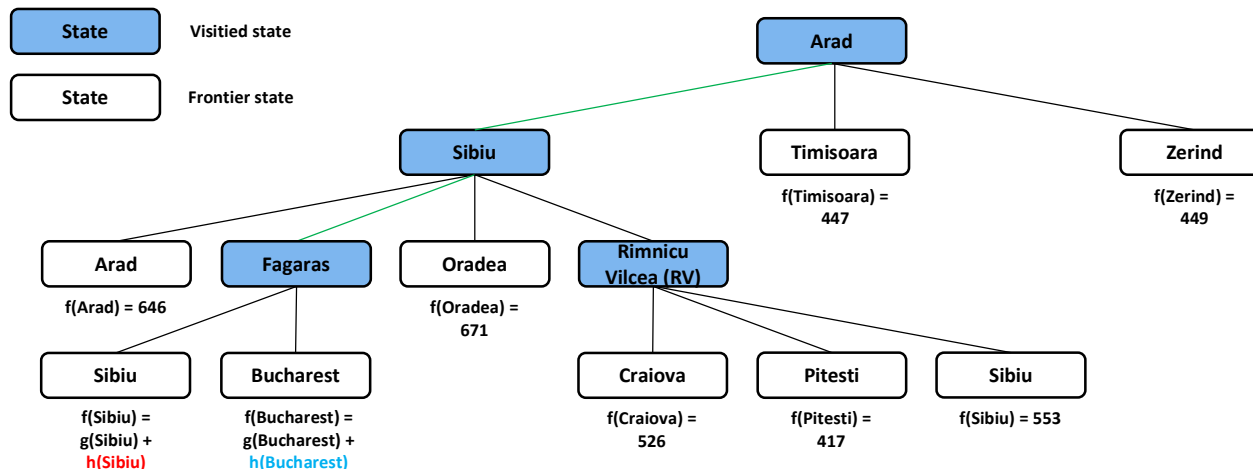


Dracula's Roadtrip: A* Search

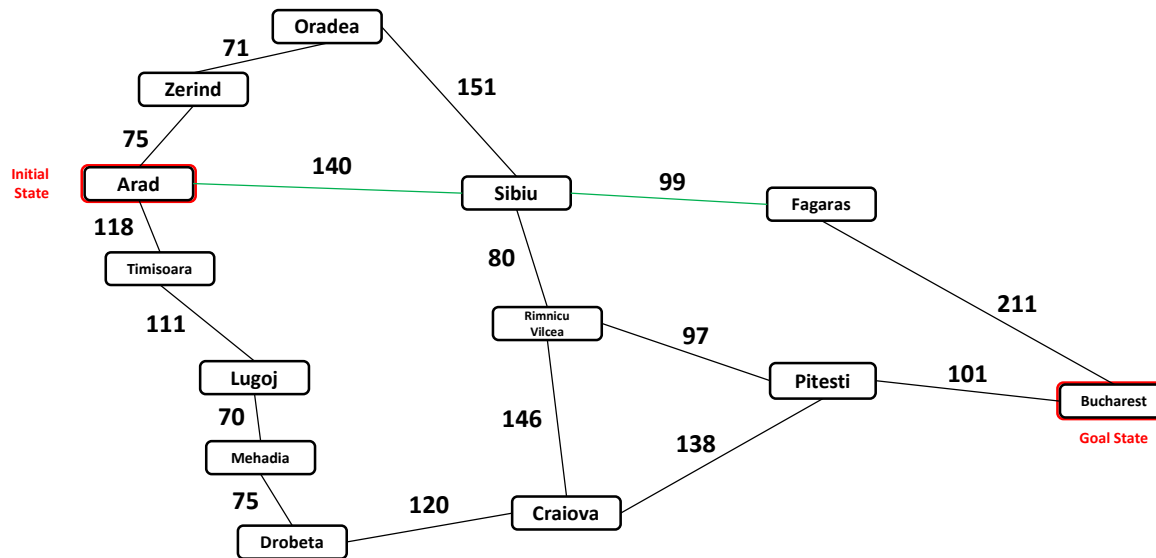


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

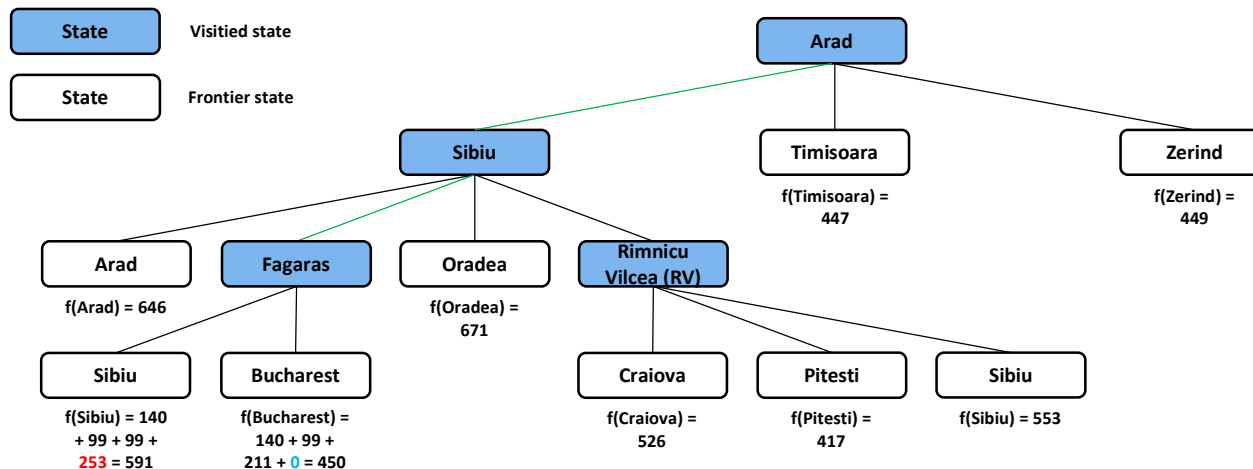


Dracula's Roadtrip: A* Search

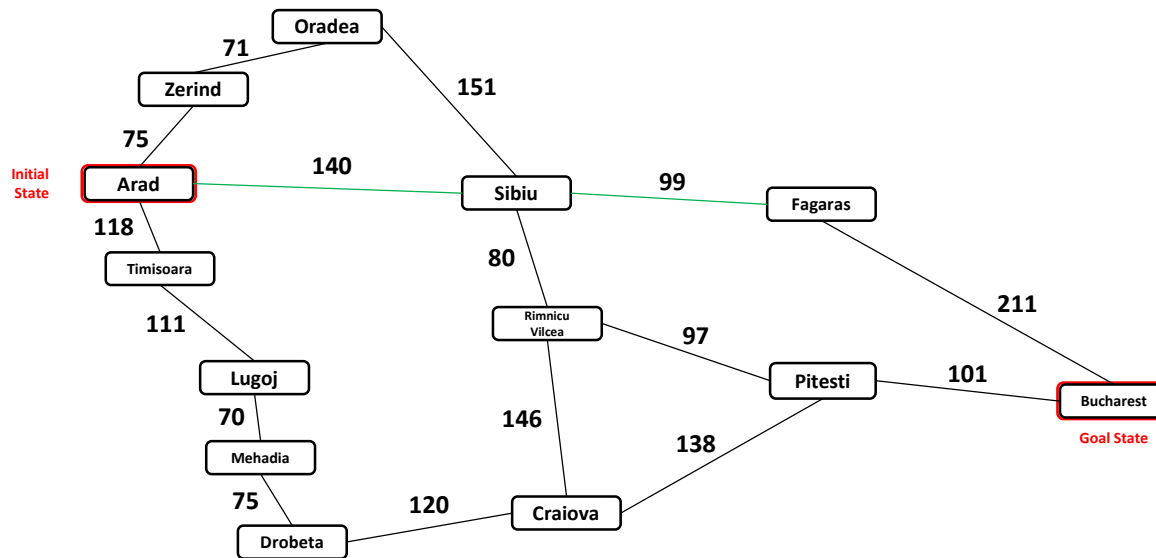


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

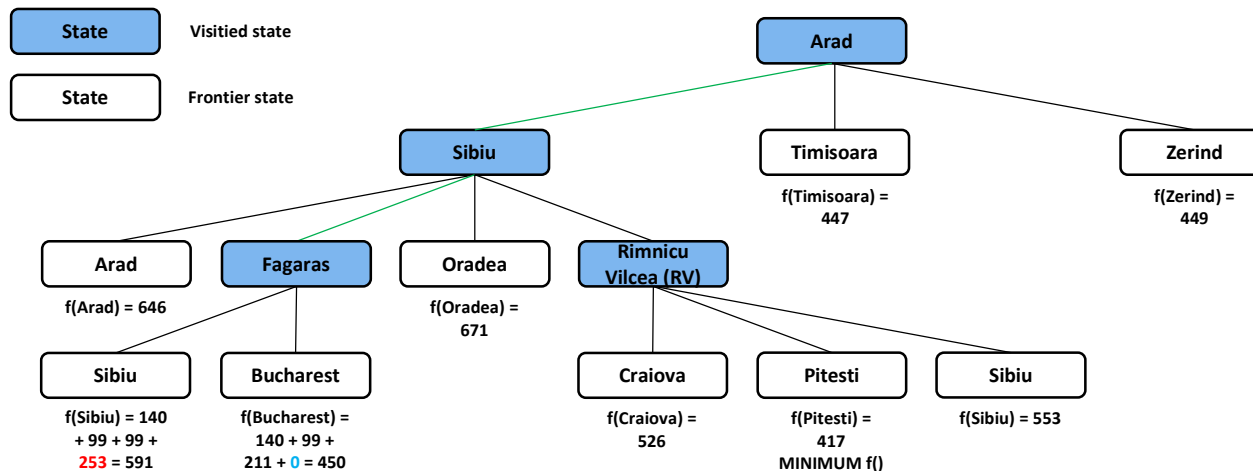


Dracula's Roadtrip: A* Search

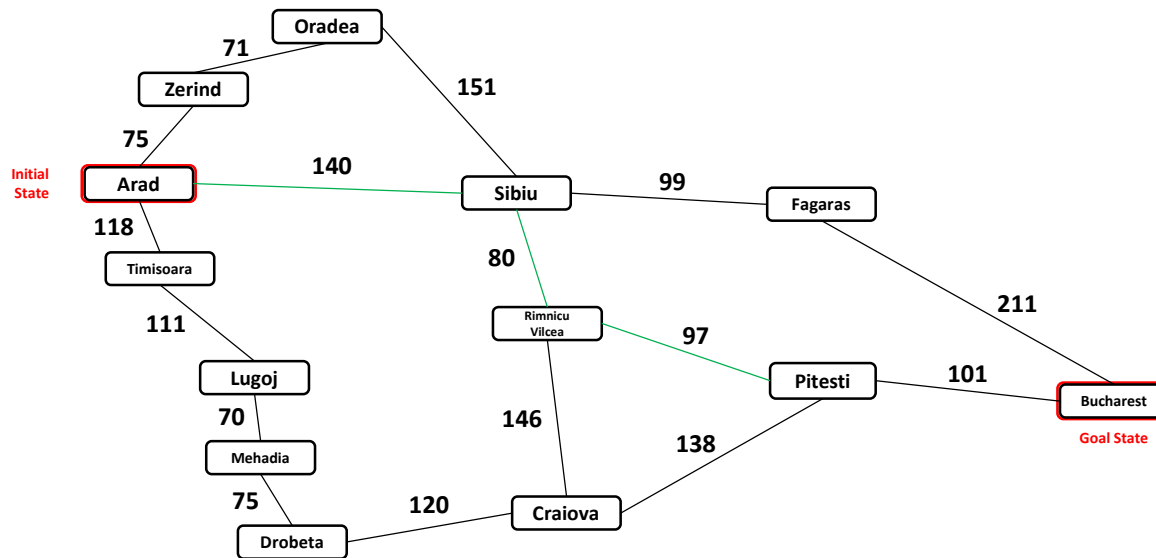


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

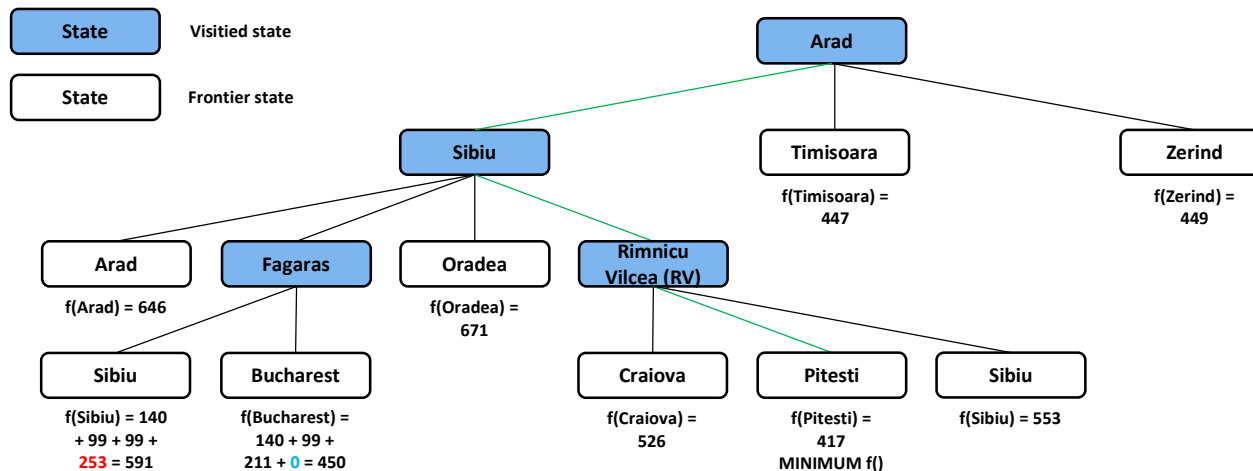


Dracula's Roadtrip: A* Search

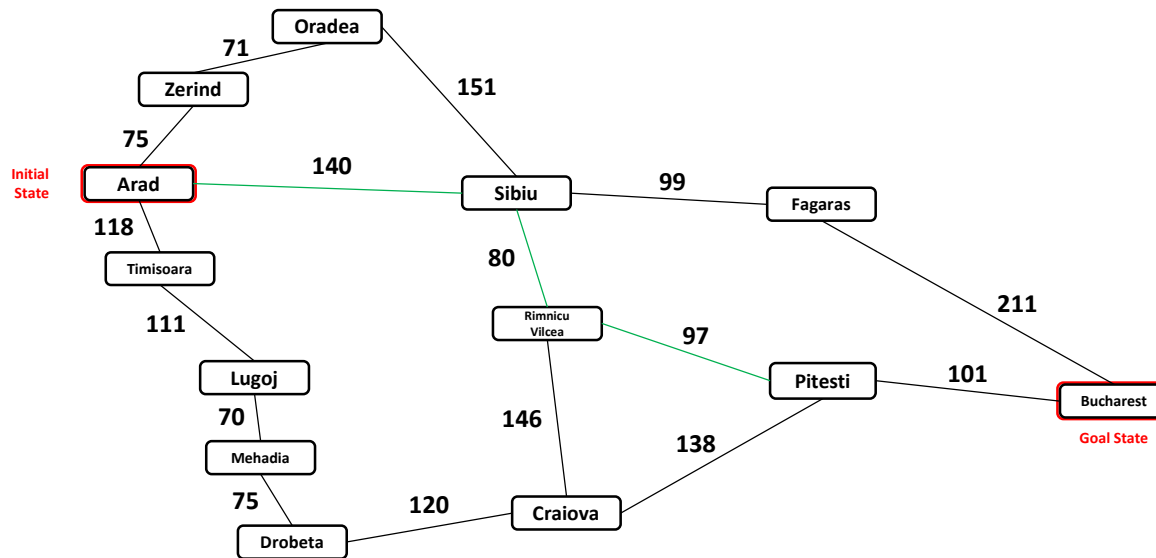


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

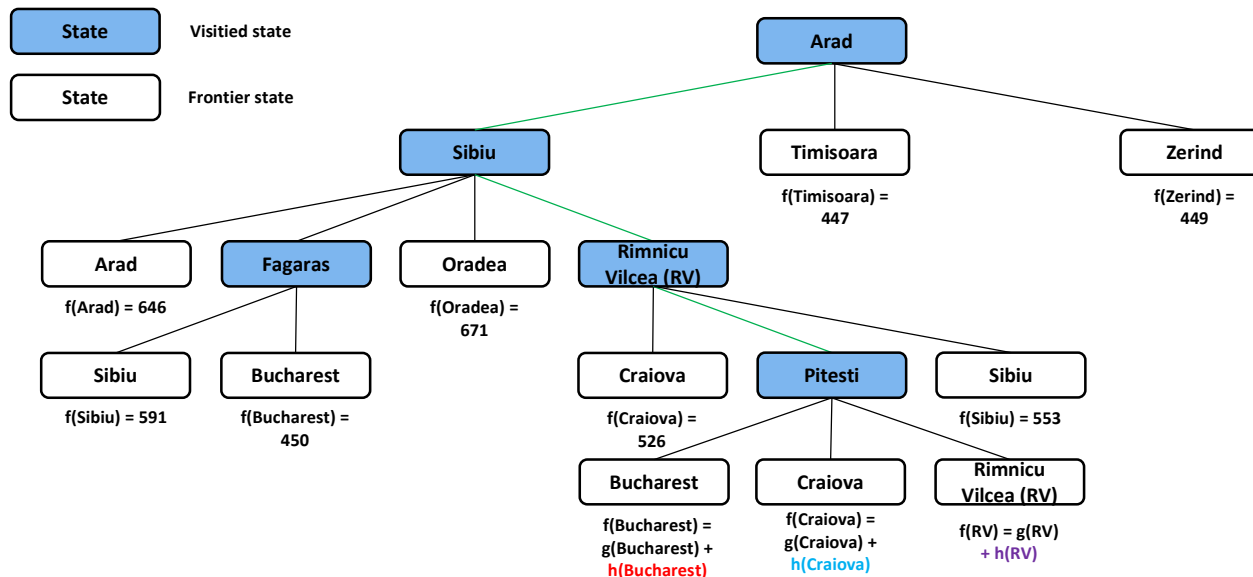


Dracula's Roadtrip: A* Search

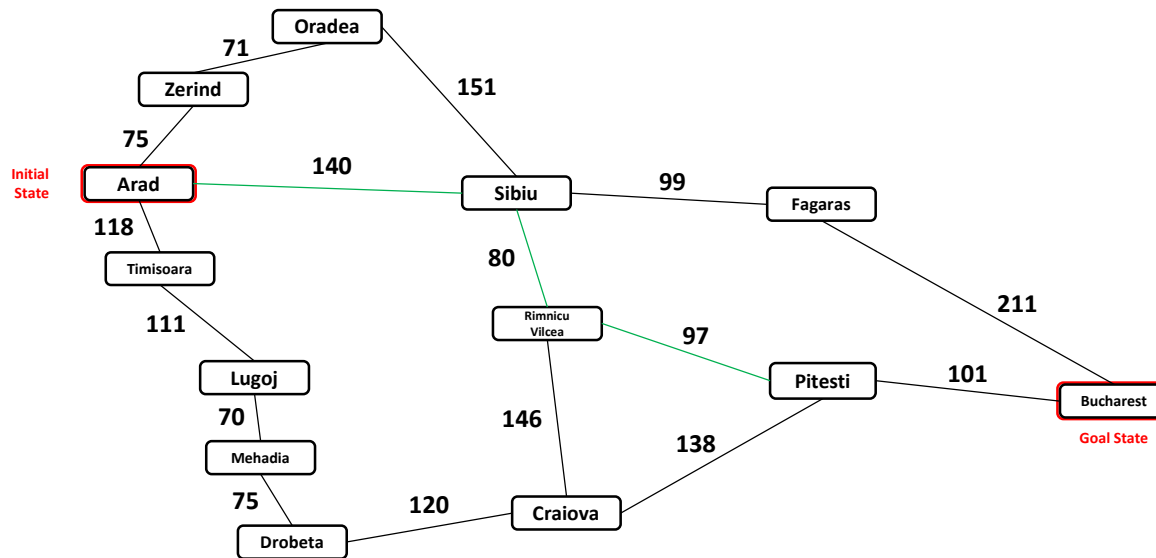


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu	
Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

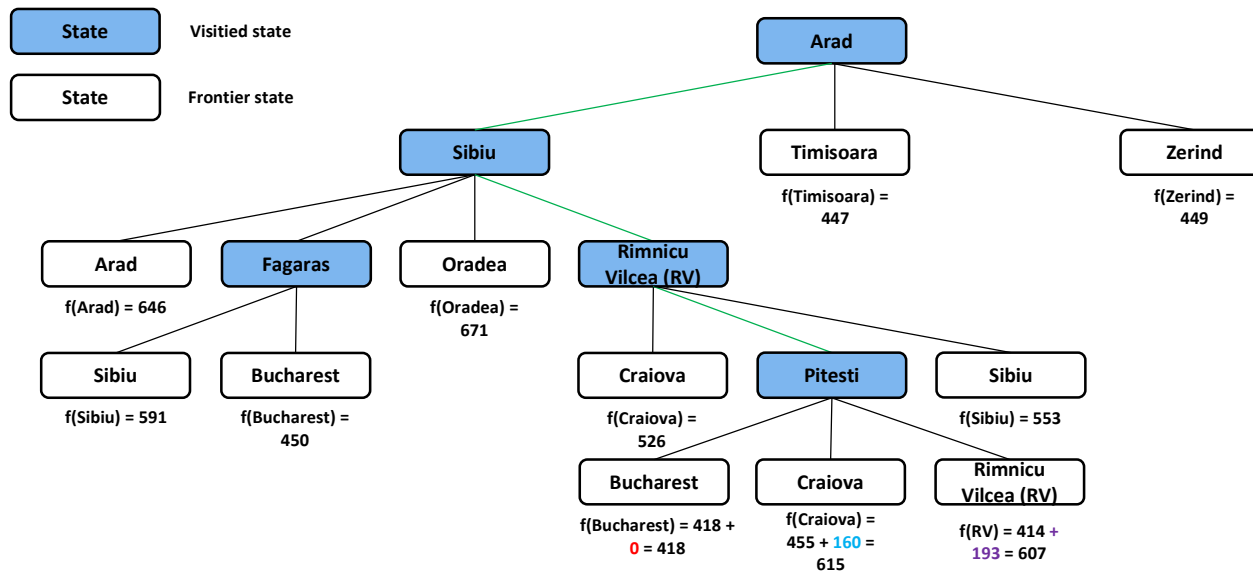


Dracula's Roadtrip: A* Search

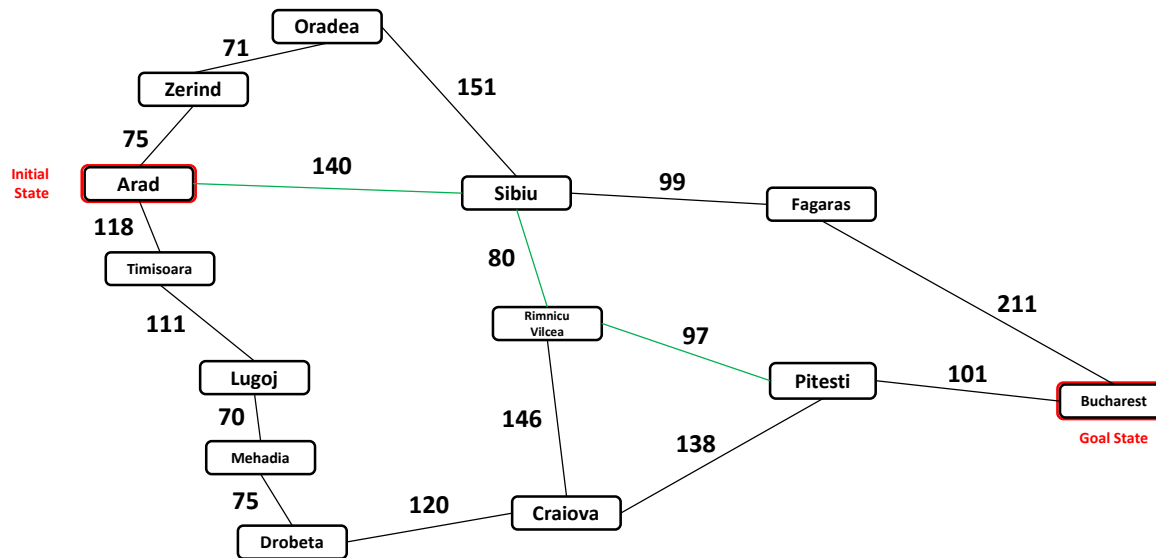


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

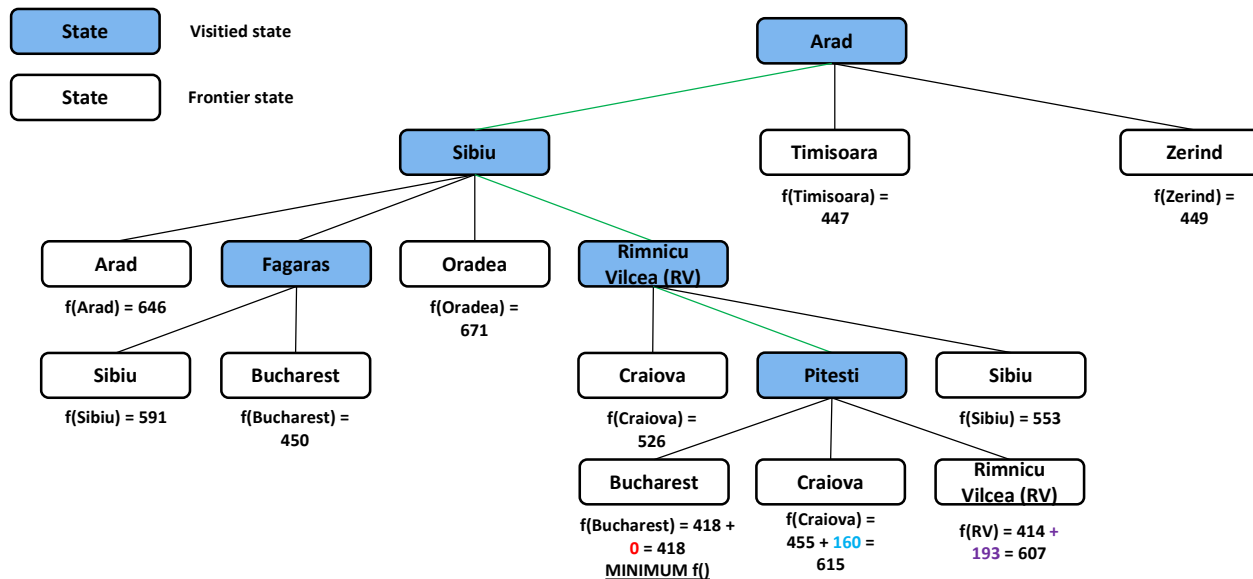


Dracula's Roadtrip: A* Search

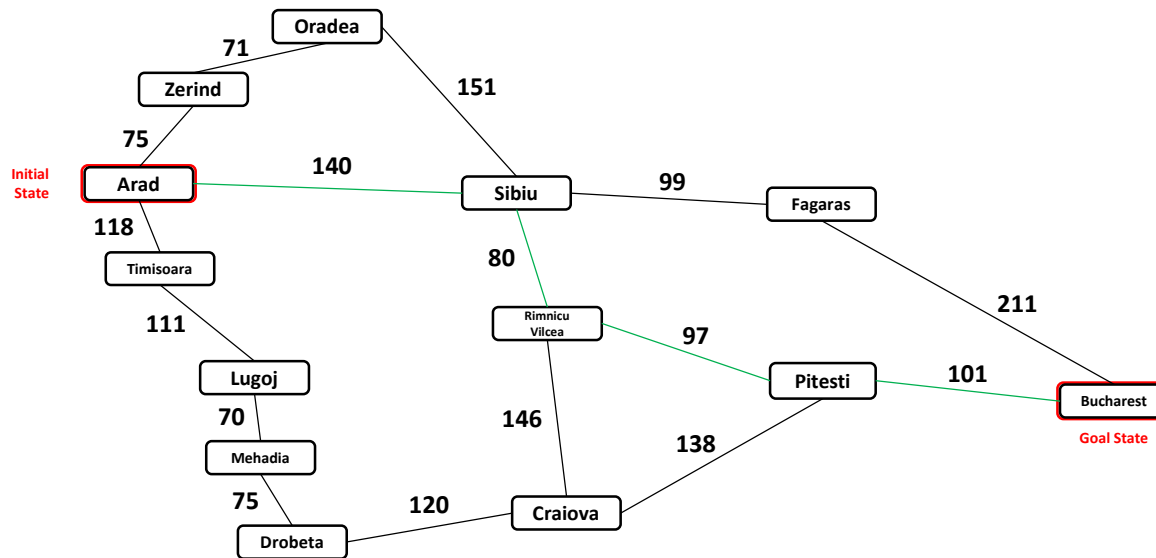


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu	
Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

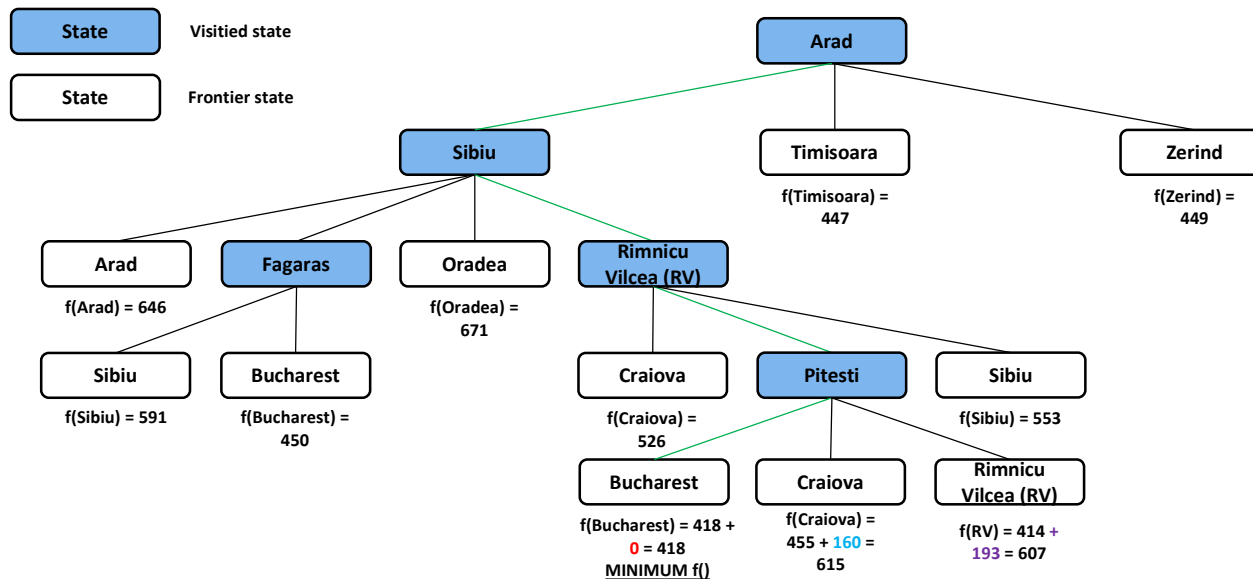


Dracula's Roadtrip: A* Search

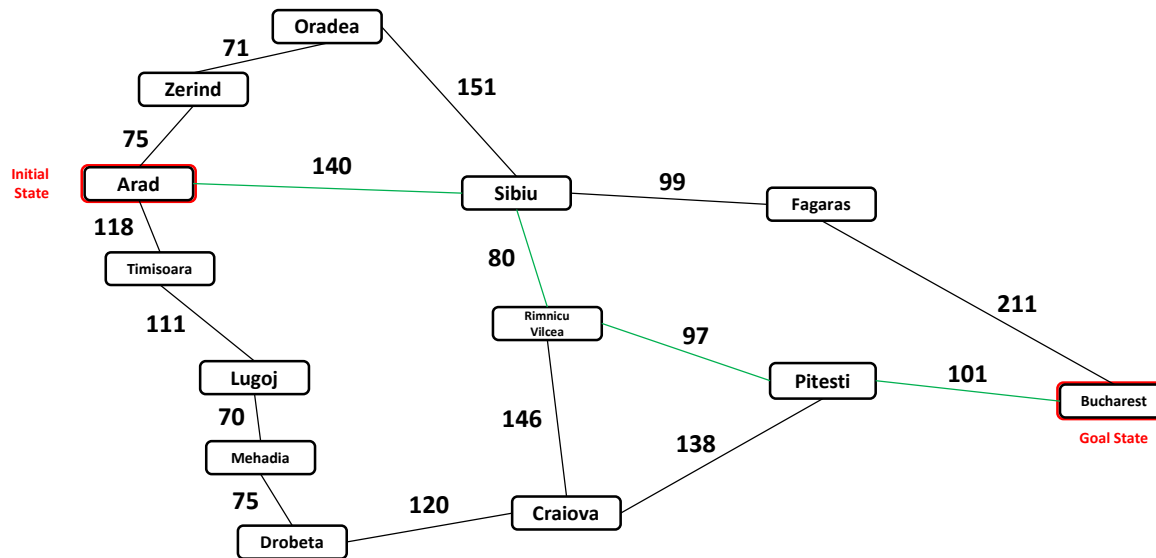


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

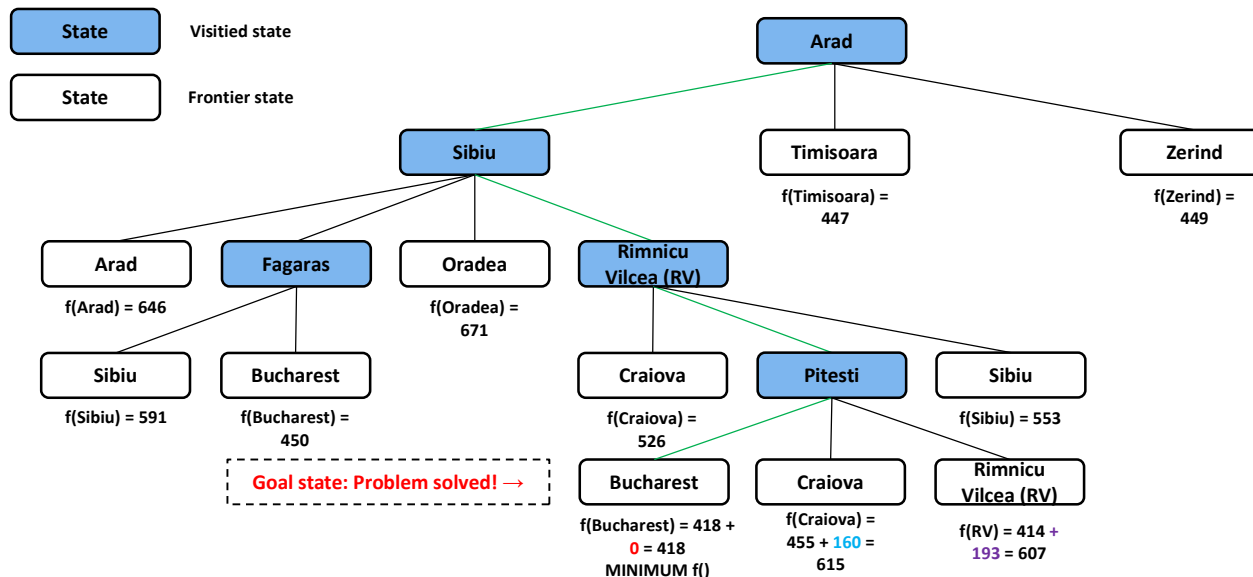


Dracula's Roadtrip: A* Search

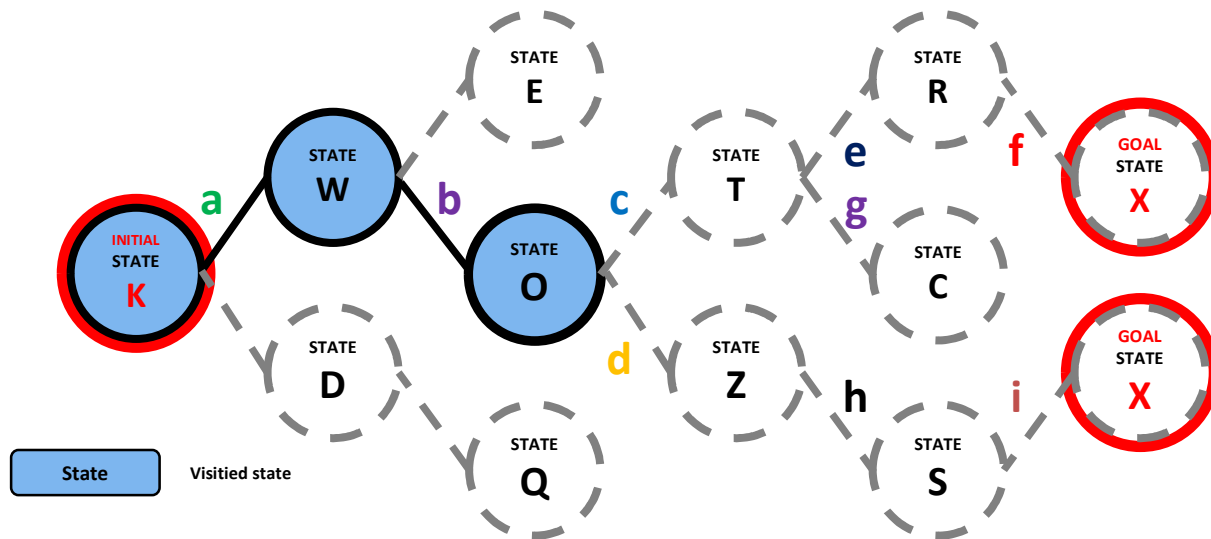


Straight-line distance to Bucharest ($h(\text{State})$):

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadi	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



Hill Climbing Search vs. A* Search



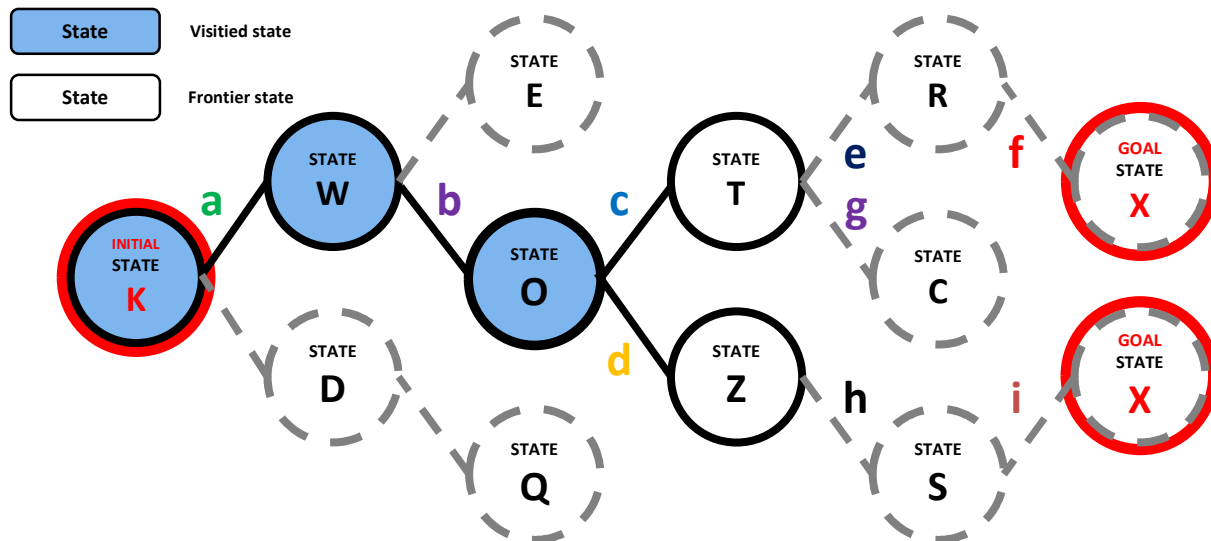
Best First Search:

Go to T or Z?

$$f(T) = c$$

$$f(Z) = d$$

Pick state with min $f()$



A* Search:

Expand T or Z?

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

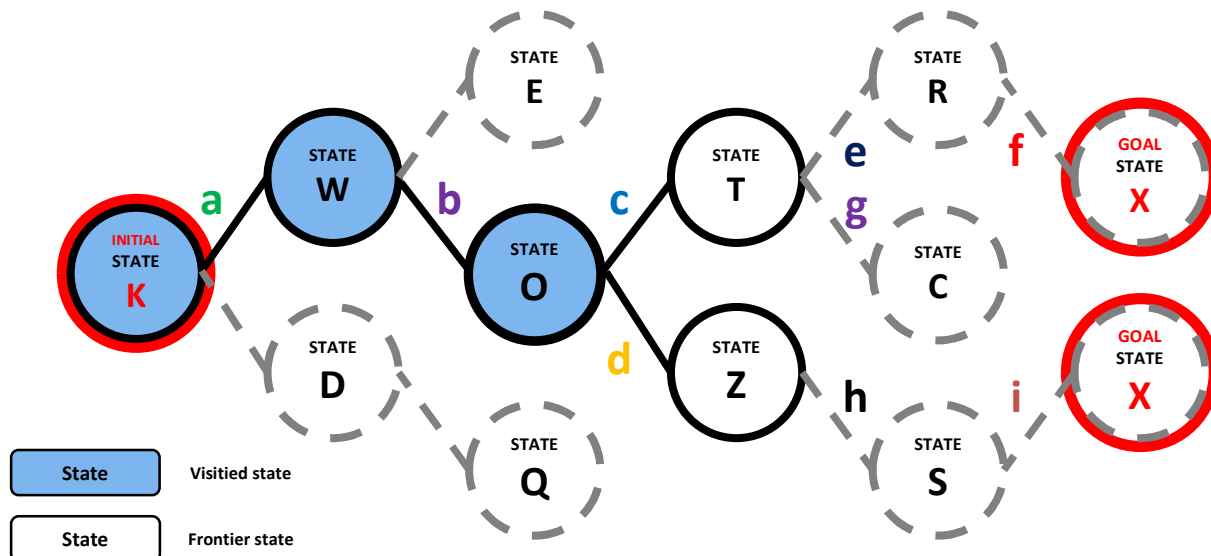
$$f(T) = a + b + c + h(T)$$

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

$$f(T) = a + b + d + h(Z)$$

Pick state with min $f()$

Greedy Best First Search vs. A* Search



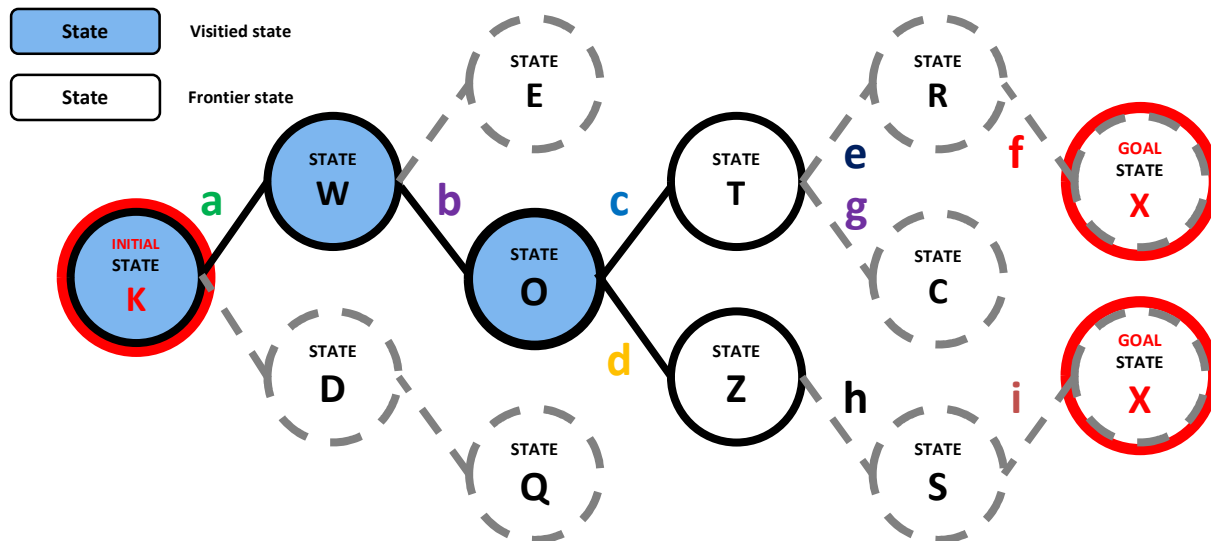
Greedy Best First:

Expand T or Z?

$$f(T) = h(T)$$

$$f(Z) = h(Z)$$

Pick state with min $f()$



A* Search:

Expand T or Z?

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

$$f(T) = a + b + c + h(T)$$

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

$$f(T) = a + b + d + h(Z)$$

Pick state with min $f()$