Artificial Intelligence

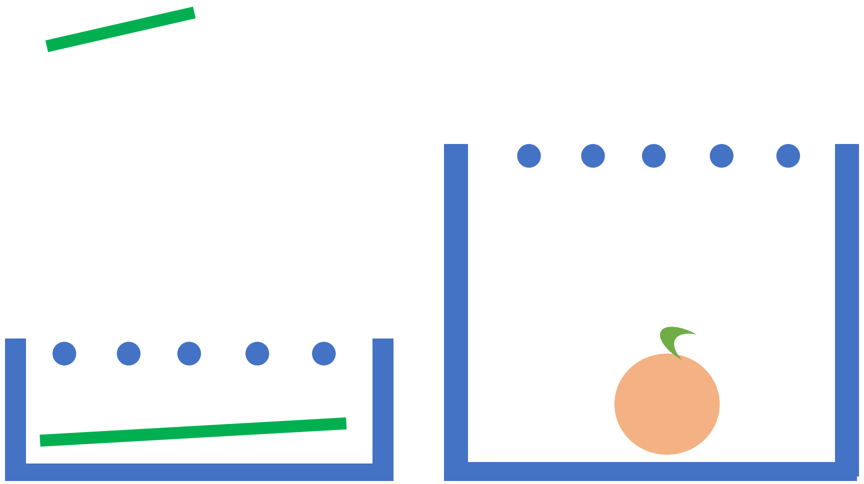
Lecture 03: Searching

Lecture Outline

- State Space
- Basic Search Algorithms
  - Breadth-first Search
  - Depth-first Search
  - Uniform Cost Search
- Describing Searches
  - Complexity
  - Completeness
  - Optimality

Reflex vs Planning

- Monkey with 2 sticks



https://www.youtube.com/watch?v=Gui3IswQ0DI

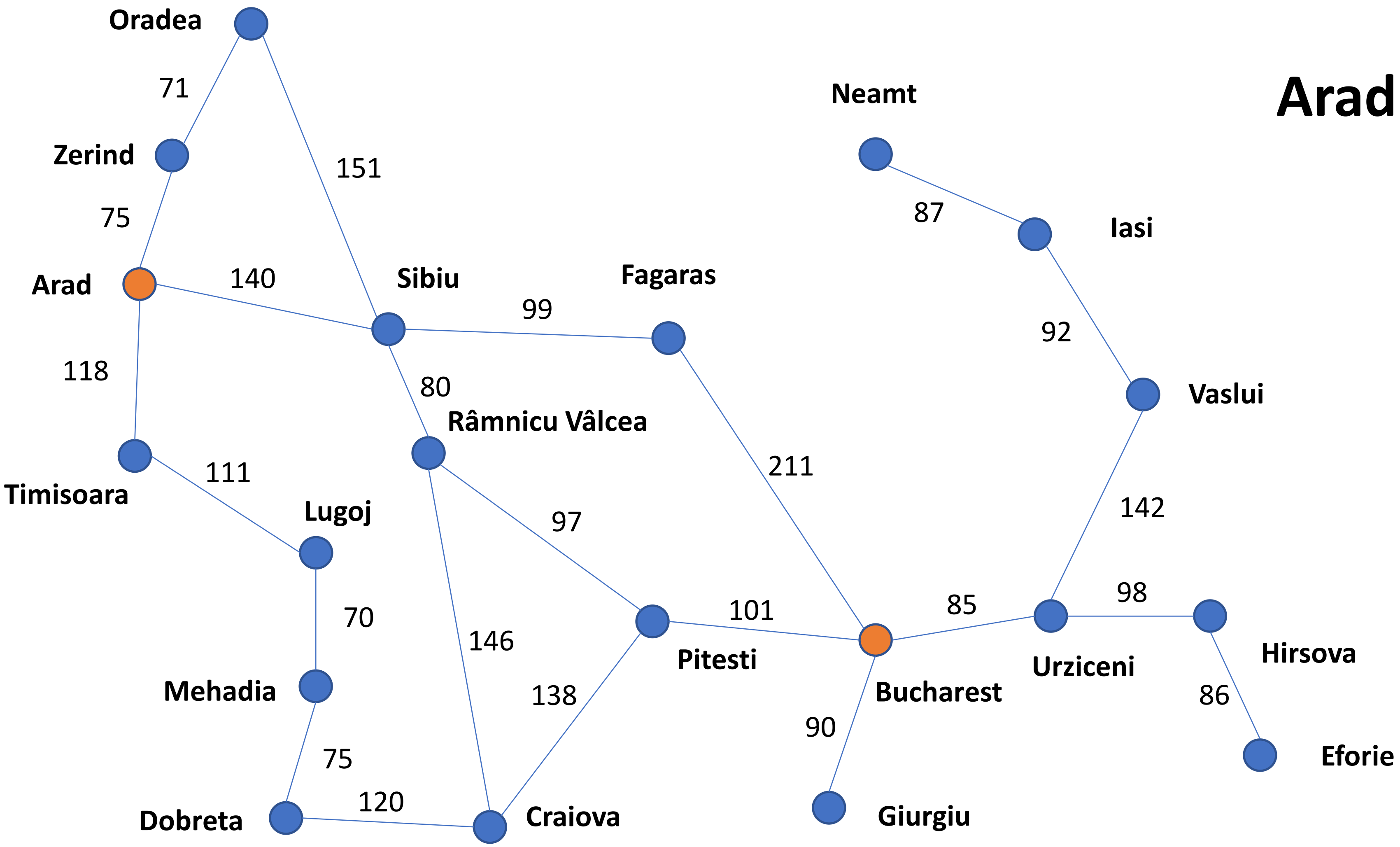
State Space

What is a state space?

The Problem!

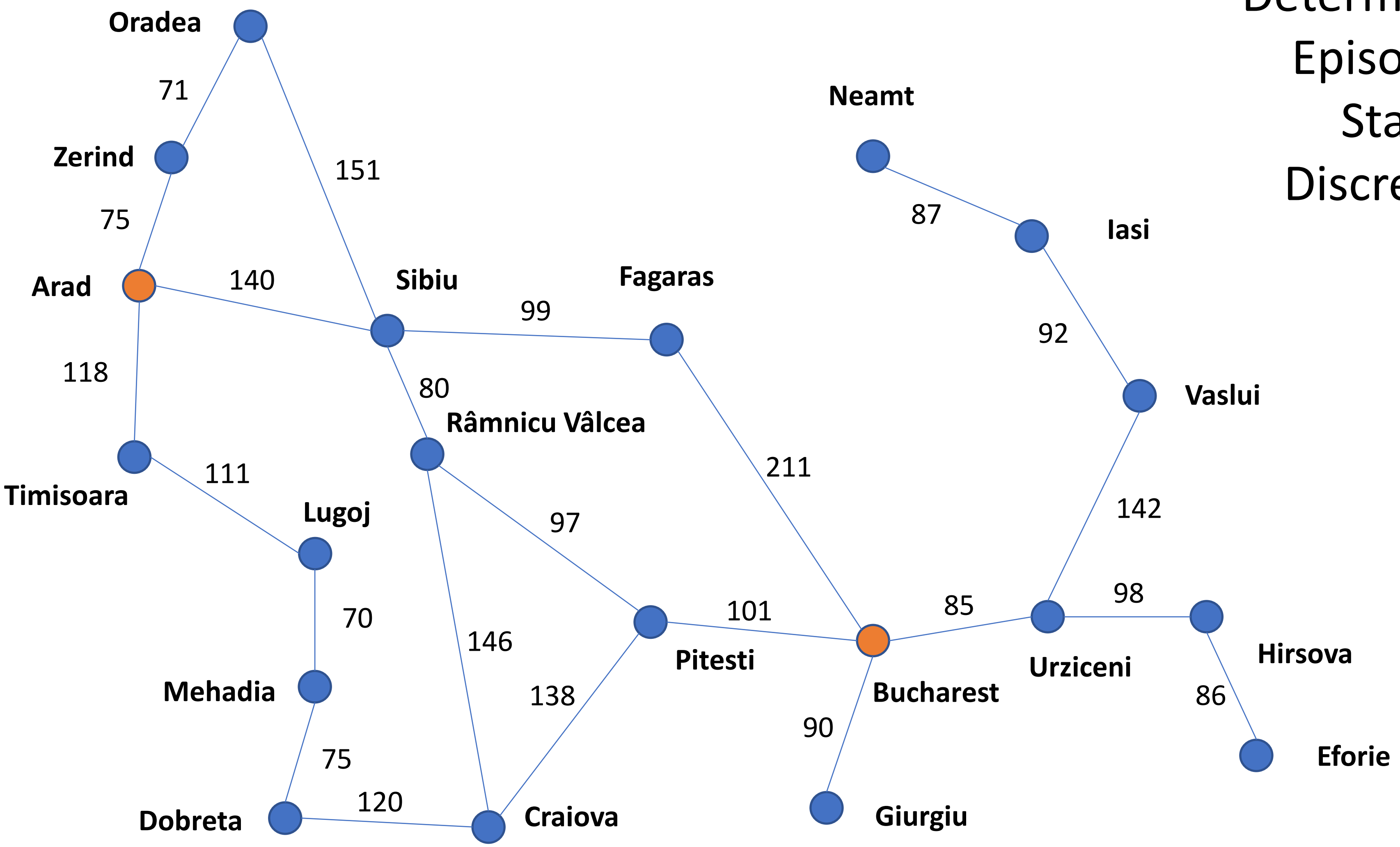
How to get from…

Arad to **Bucharest**!



What kind of Problem?

Fully Observable vs Partially Observable
Deterministic vs Stochastic
Episodic vs Sequential
Static vs Dynamic
Discrete vs Continuous



What kind of Problem?

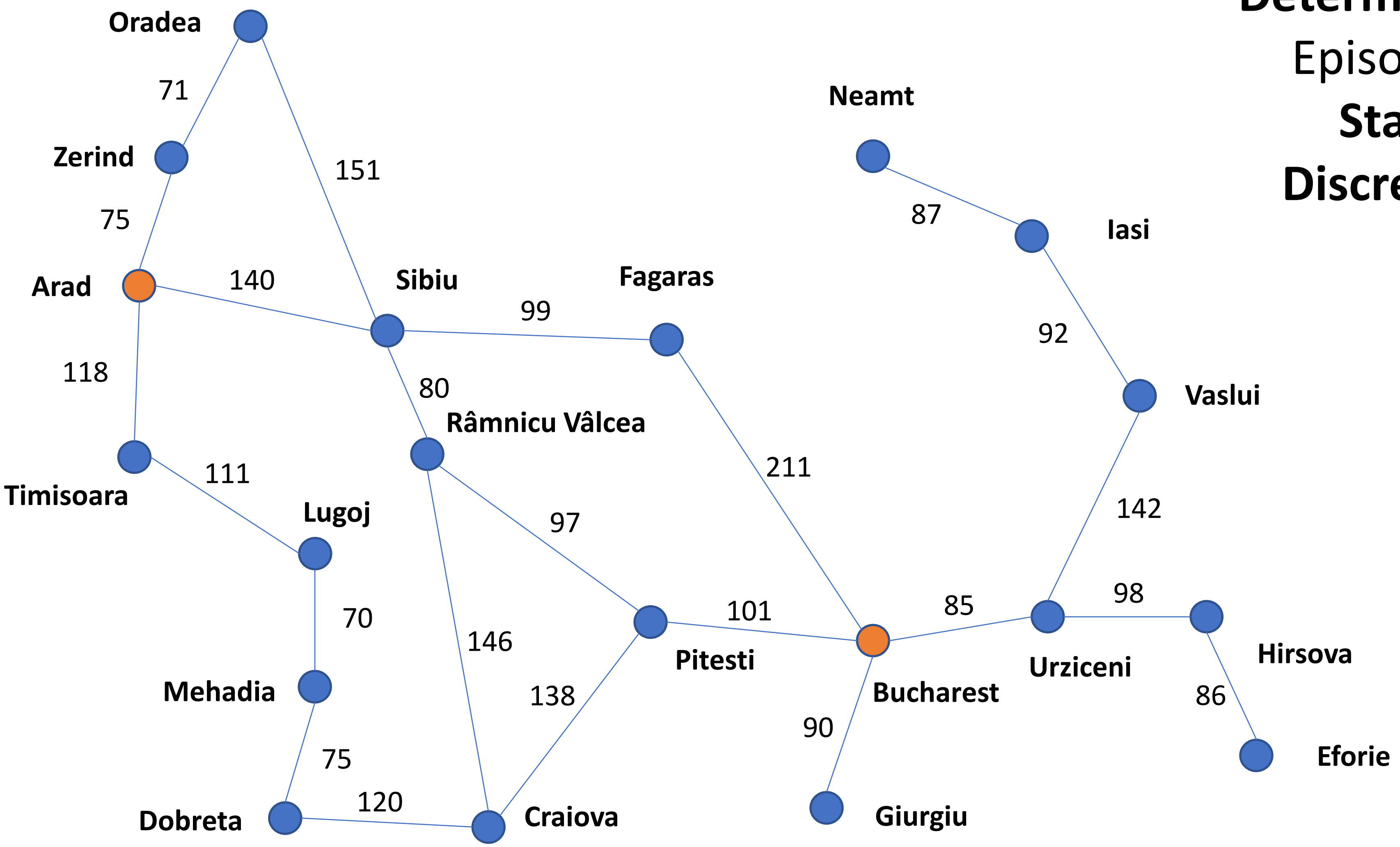
Fully Observable vs ~~Partially Observable~~

Deterministic vs ~~Stochastic~~

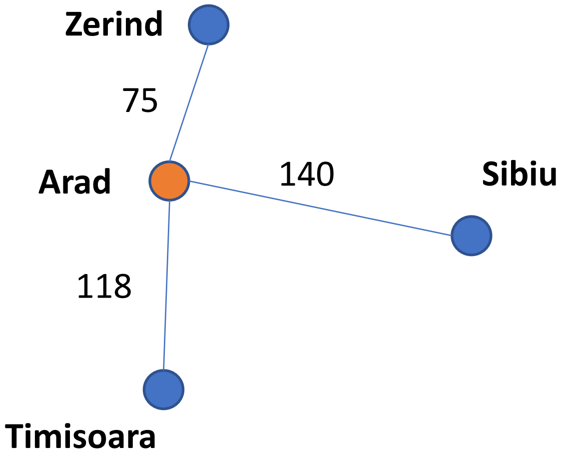
Episodic vs Sequential

Static vs ~~Dynamic~~

Discrete vs ~~Continuous~~



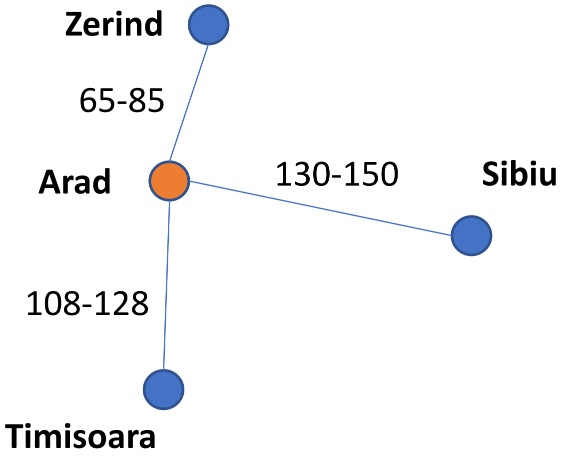
How could we change this problem?



What if the state-space is unknown?

→ **Exploration** problem

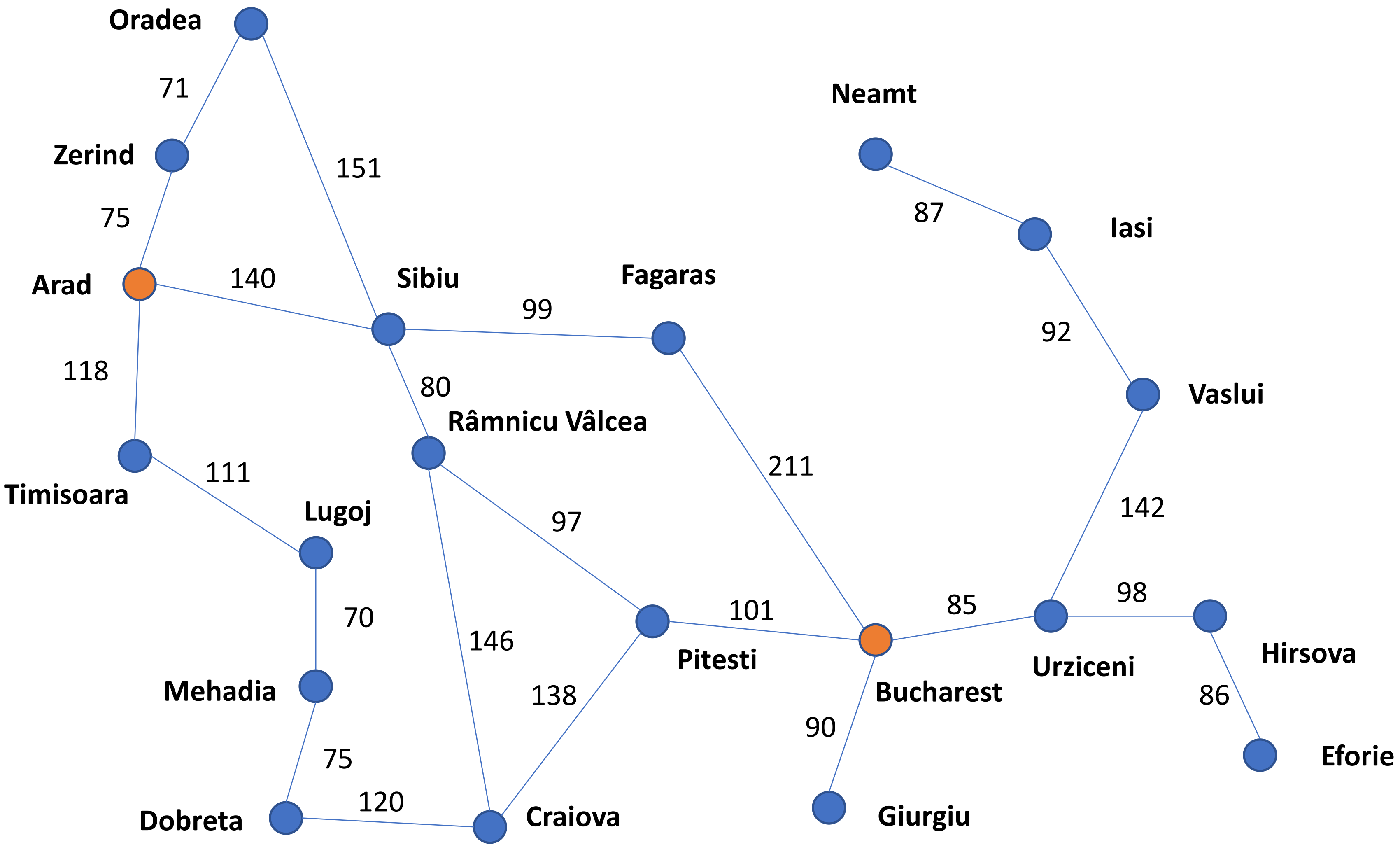
How could we change this problem?



What if the distances are times… or expected times?

→ **Contingency** problem

What kind of Problem?



Fully Observable
Deterministic

=> Search

Q: Search what?

A: State Space

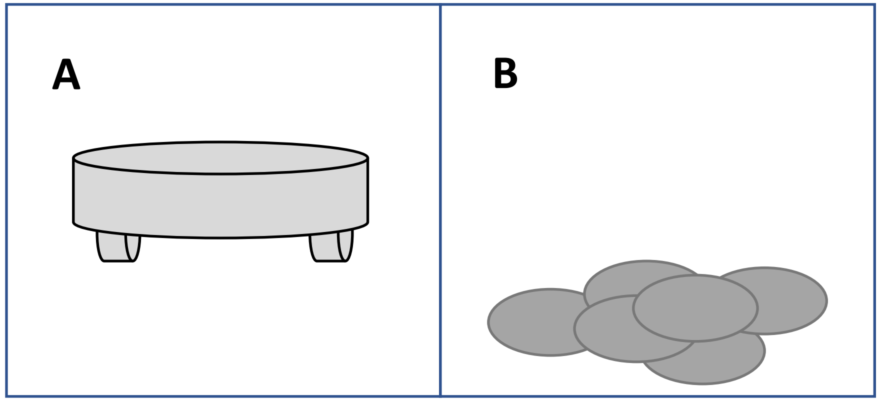
Example #1: Vacuum Cleaner

Percepts:

- Location (A, B)
- Contents (Clean, Dirty)

Actions:

- Left, Right, Suck, NoOp (do nothing)



Agent Function

Percept	**Action**
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
…	
[A, Dirty], [A, Clean]	Right
[A, Clean], [A, Dirty]	Suck

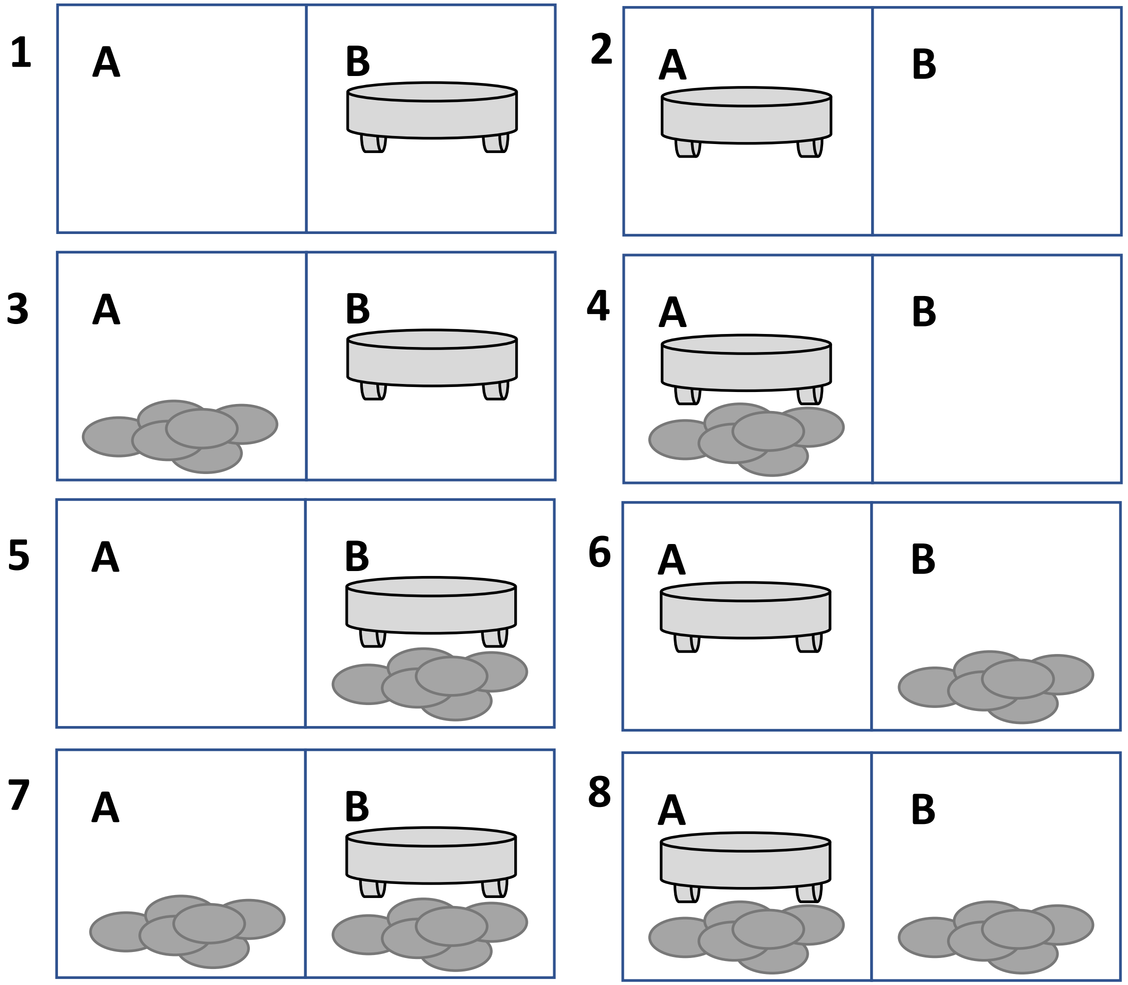
Vacuum Cleaner: States

I want to play a game.

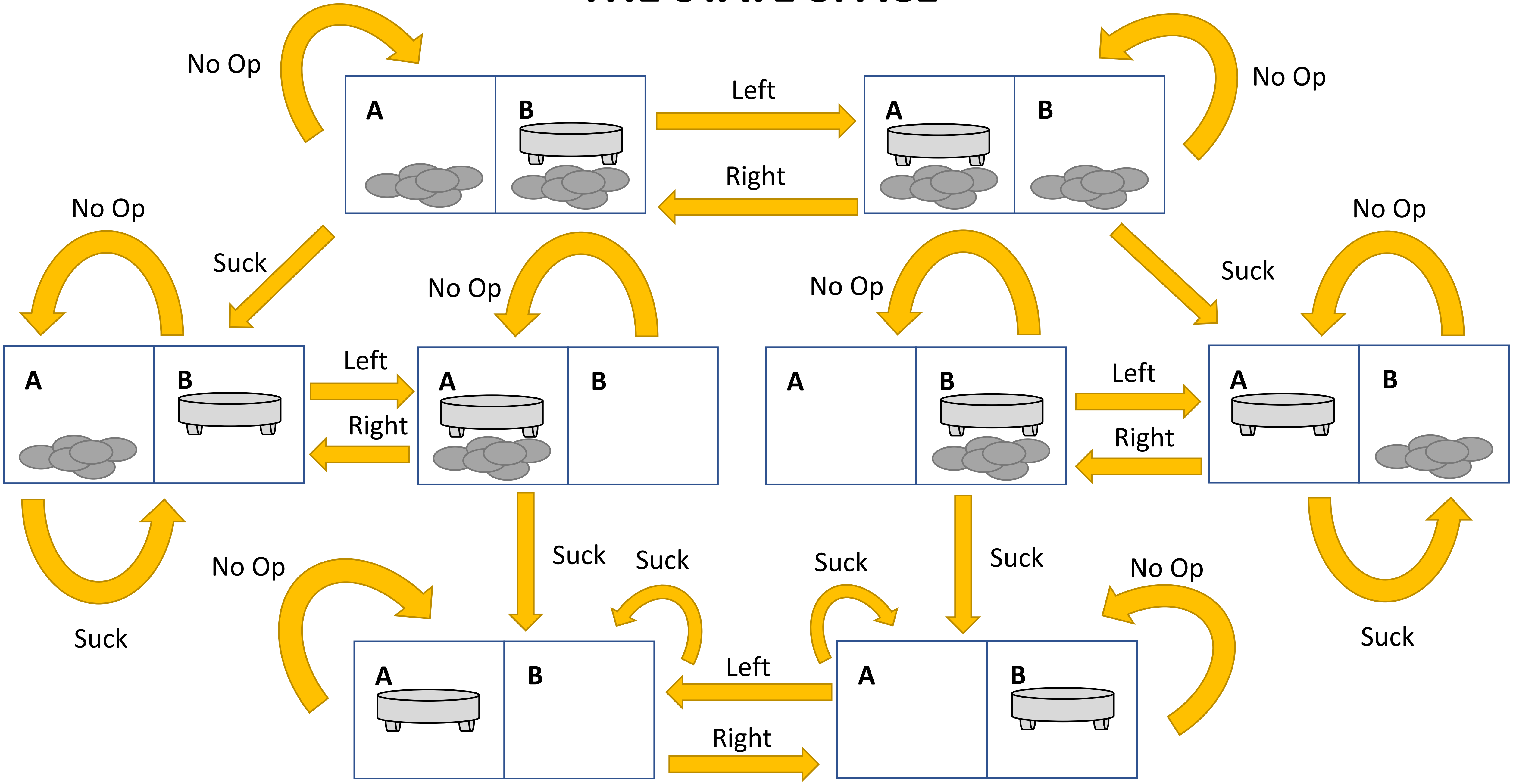
Goal: Clean Room

Start: State 7

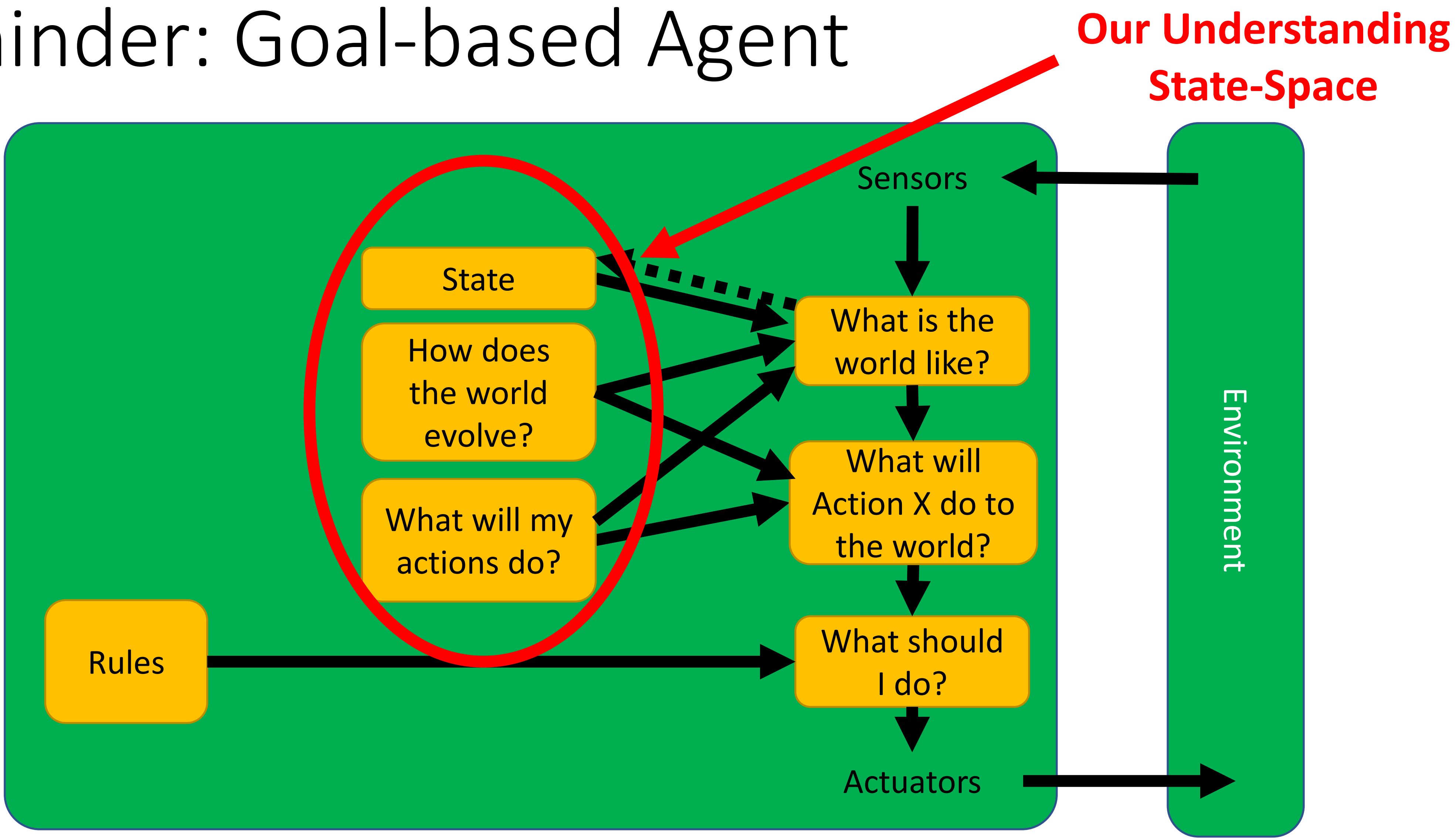
ALL POSSIBLE STATES



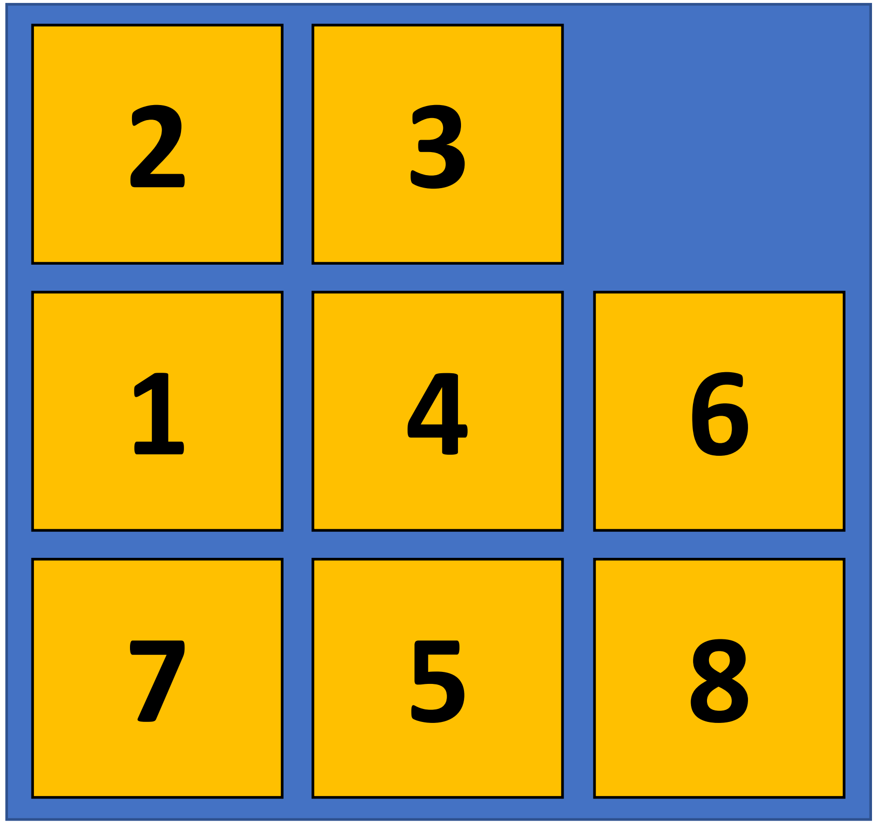
THE STATE SPACE

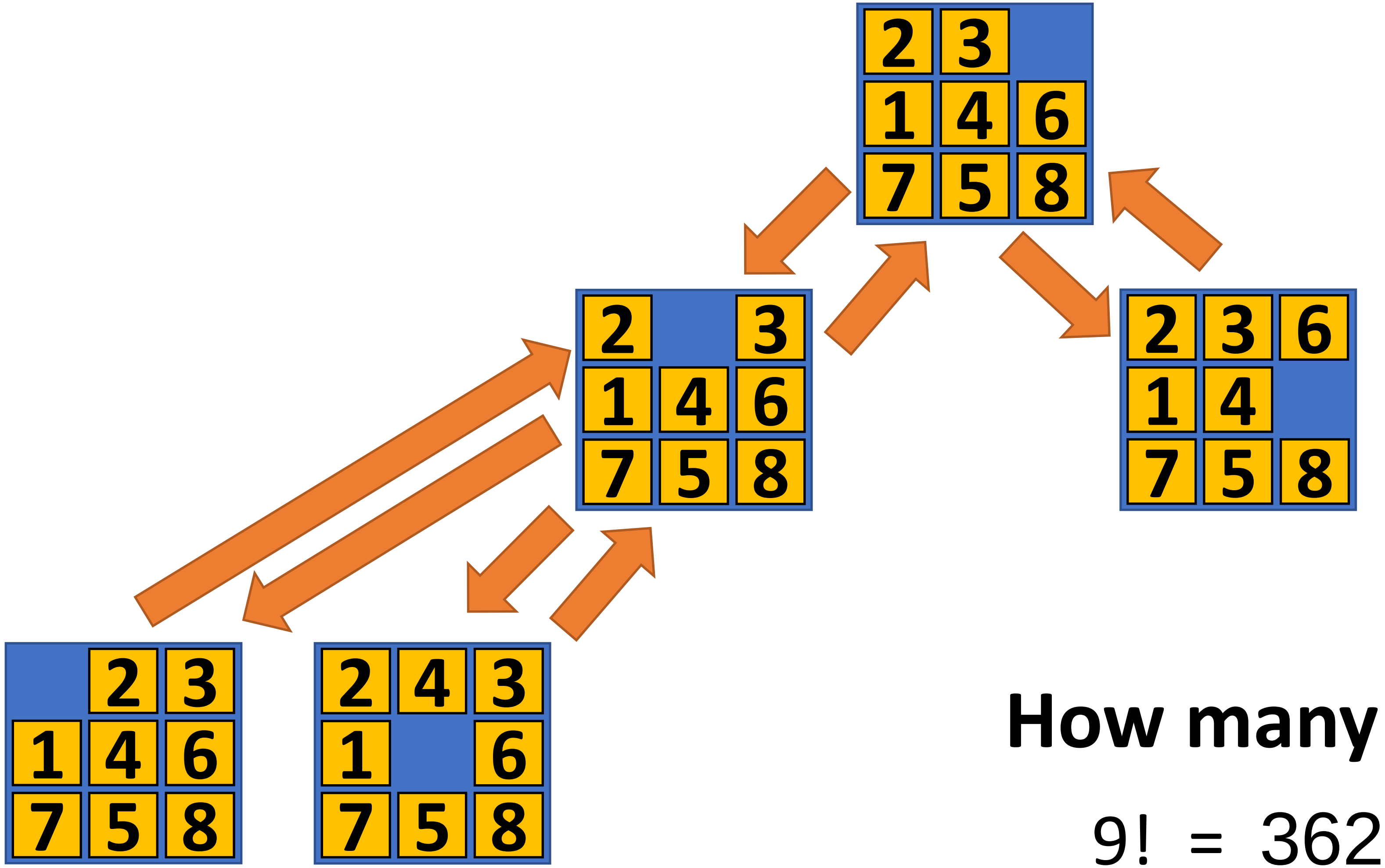


Reminder: Goal-based Agent



Example #2: 8-Puzzle





How many states?

$9! = 362880$

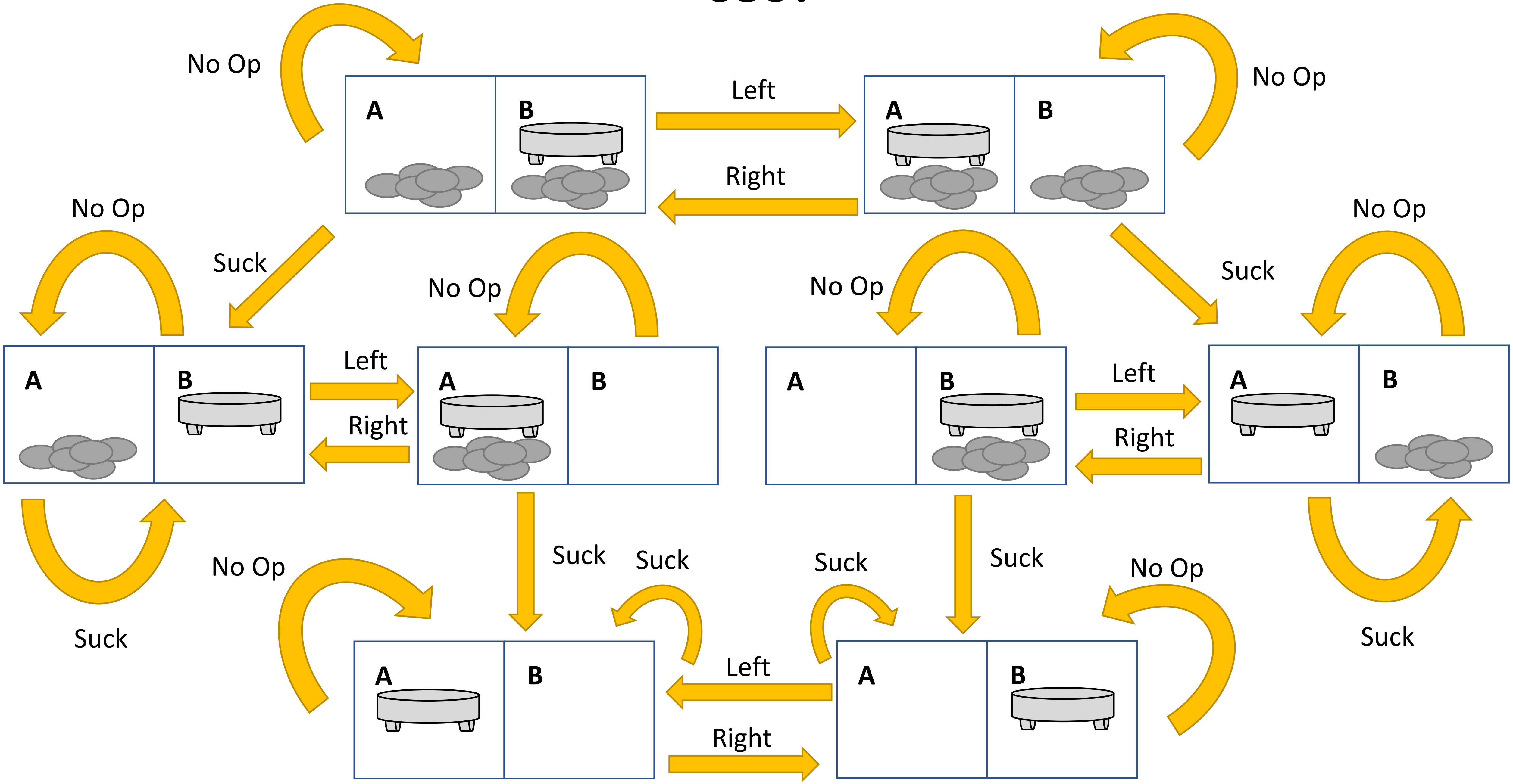
Cost

In our vacuum example…

If none of the operations had any cost, would the order in which we did the ‘cleaning matter’?

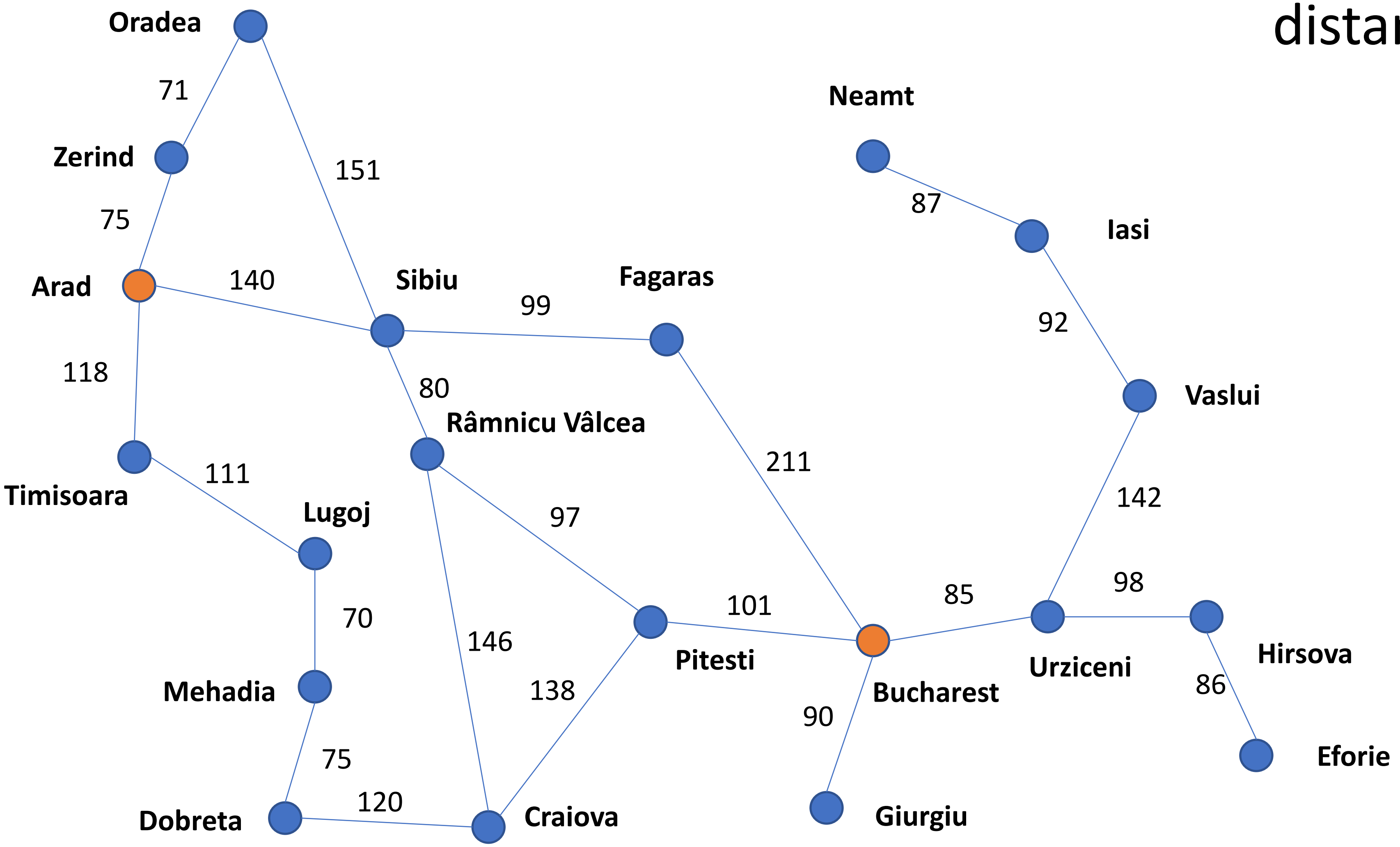
For this enterprise to make sense... We need to have a cost to our operations

COST

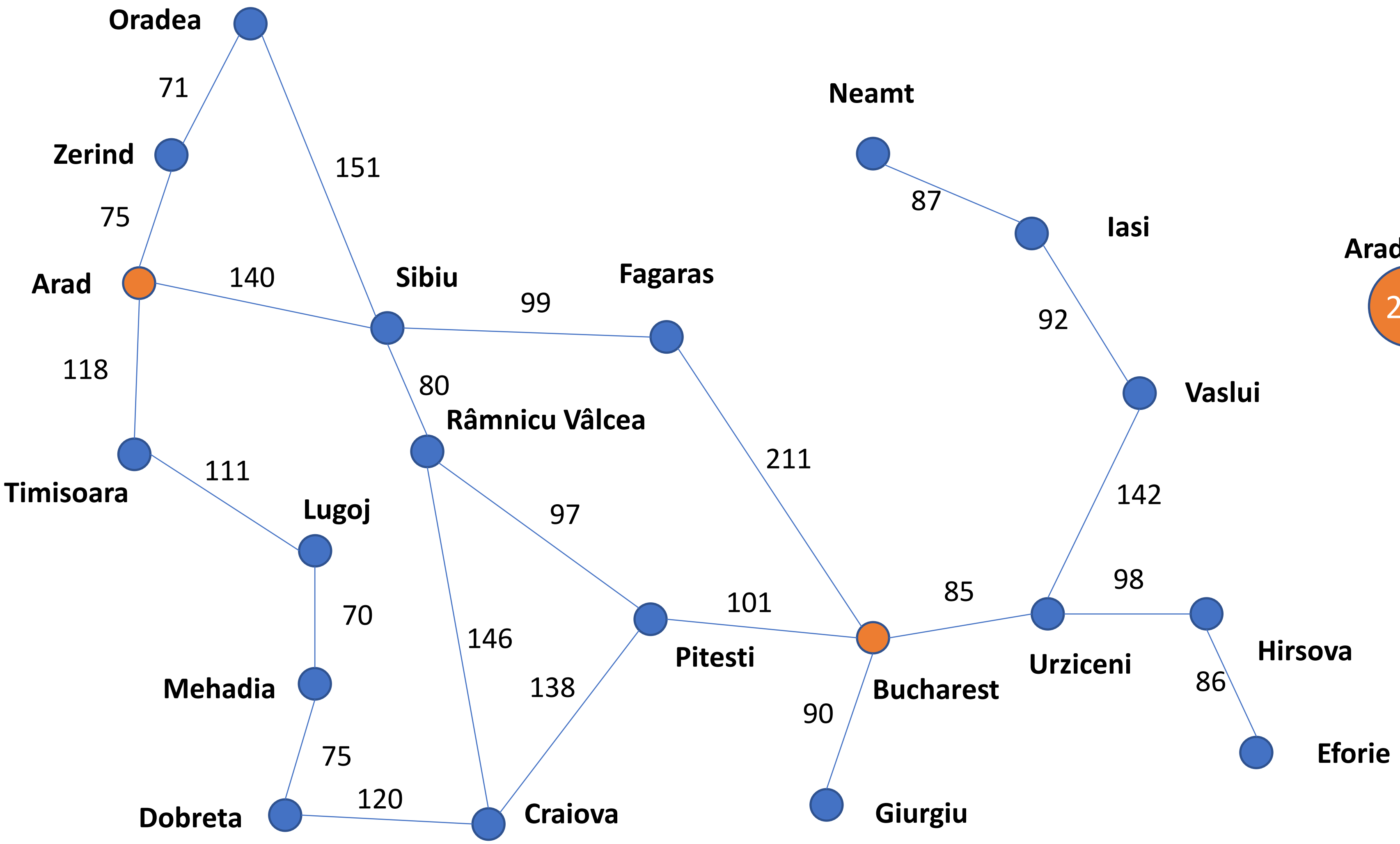

No Op
A
B
Left
Right
A
B
No Op
No Op
Suck
No Op
No Op
Suck
No Op
A
B
Left
Right
A
B
A
B
Left
Right
A
B
Suck
Suck
Suck
Suck
Suck
No Op
No Op
Suck
A
B
Left
Right
A
B

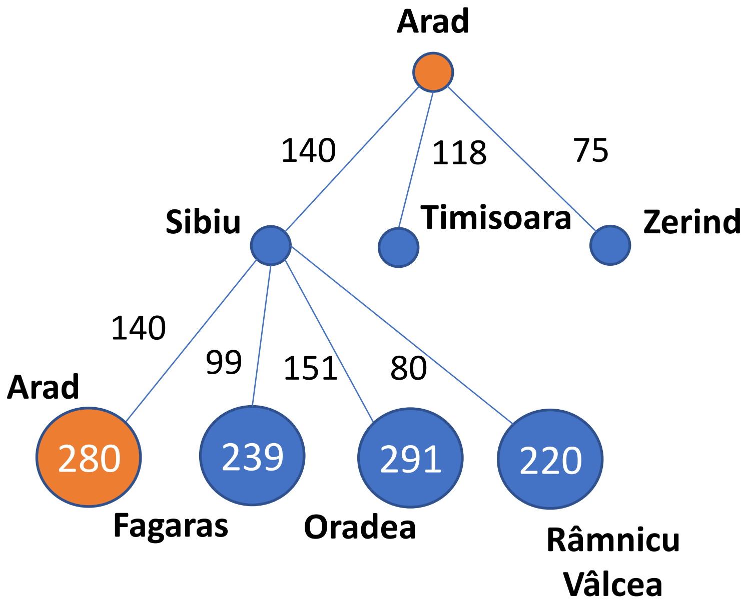
State Space with some Searching

The cost is the road distance!



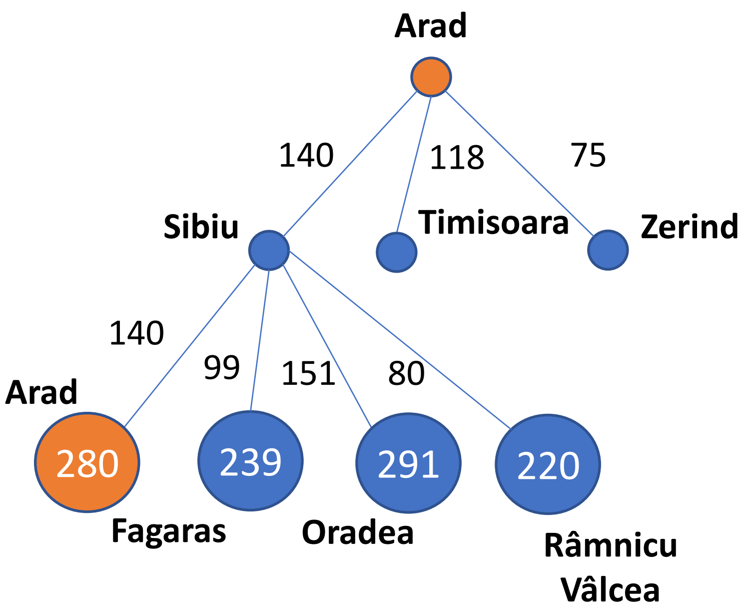
State Space with some Searching





Where to next?

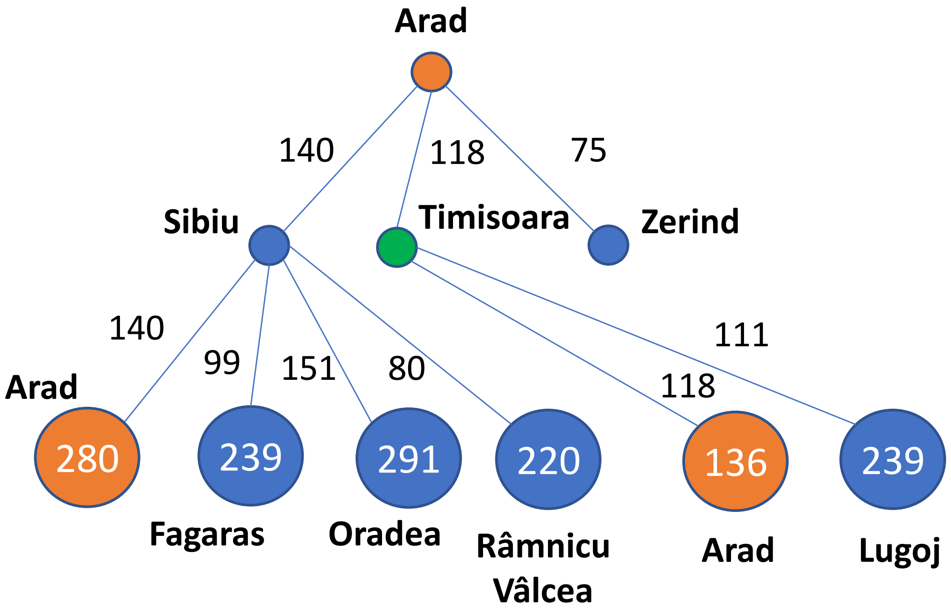
The Basic Searches (BFS, DFS)



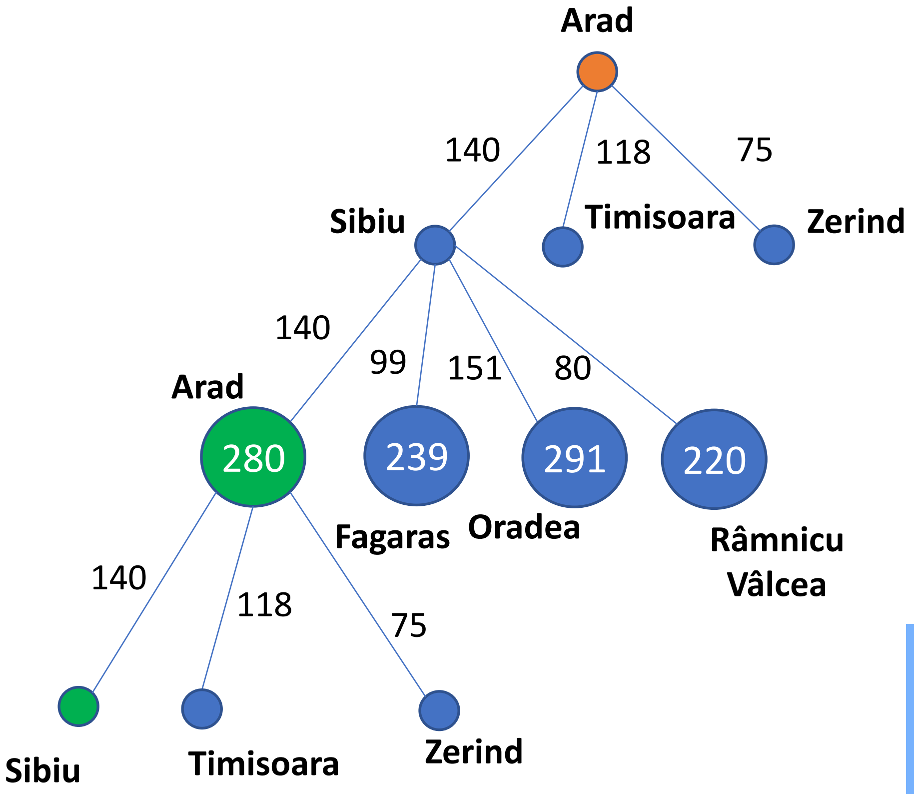
We will search from the Fringe… which is:

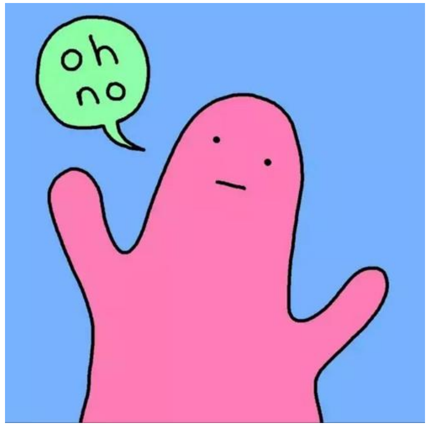
- Arad
- Fagaras
- Oradea
- Râmnicu Vâlcea
- Timisoara
- Zerind

Breadth First Search vs Depth First Search

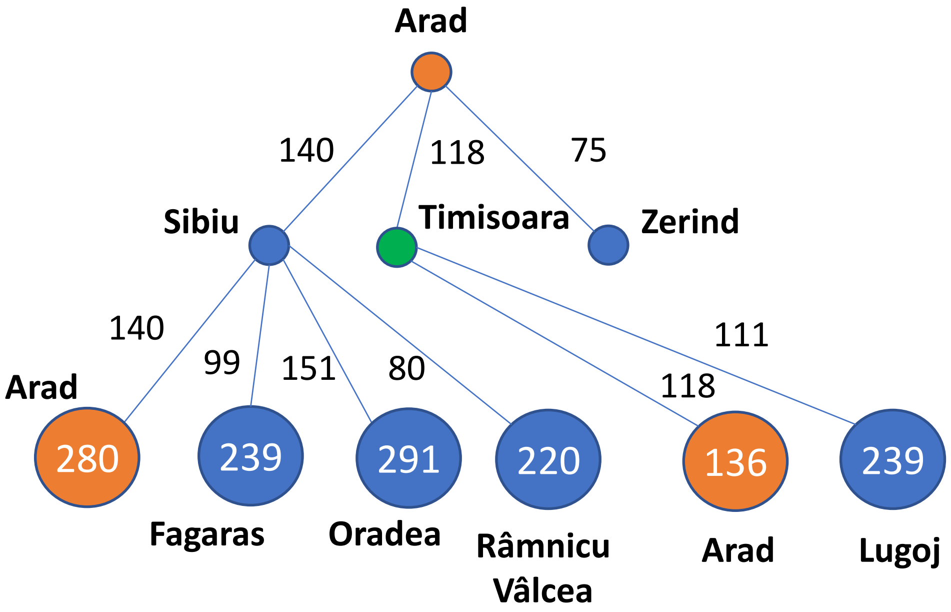


Arad
Sibiu
Timisoara
Zerind
Arad
Fagaras
Oradea
Râmnicu Vâlcea

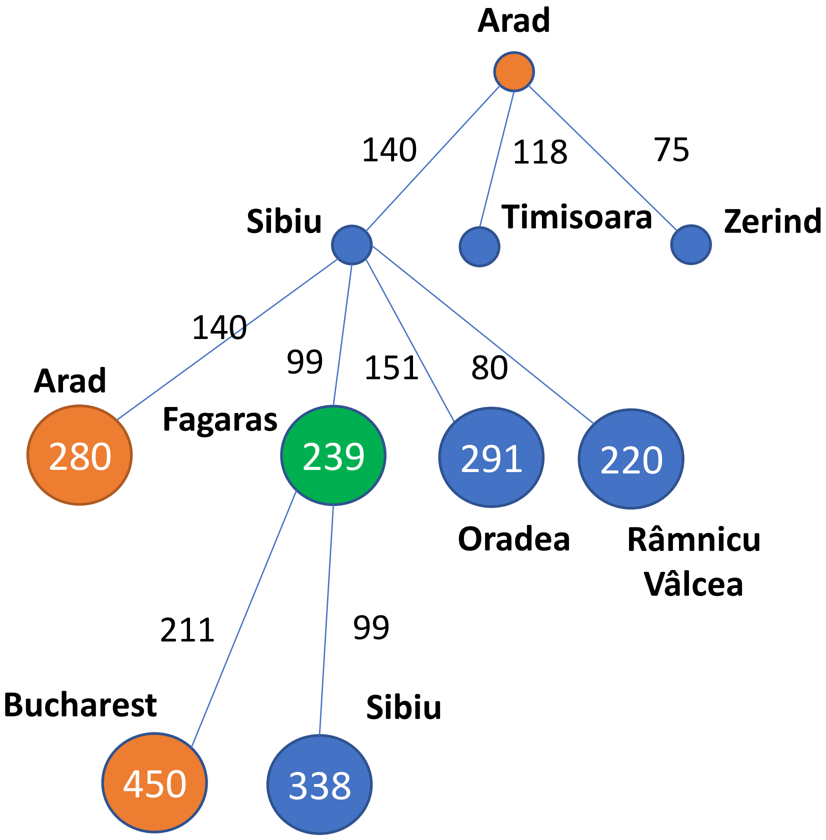




Breadth First Search vs Depth First Search

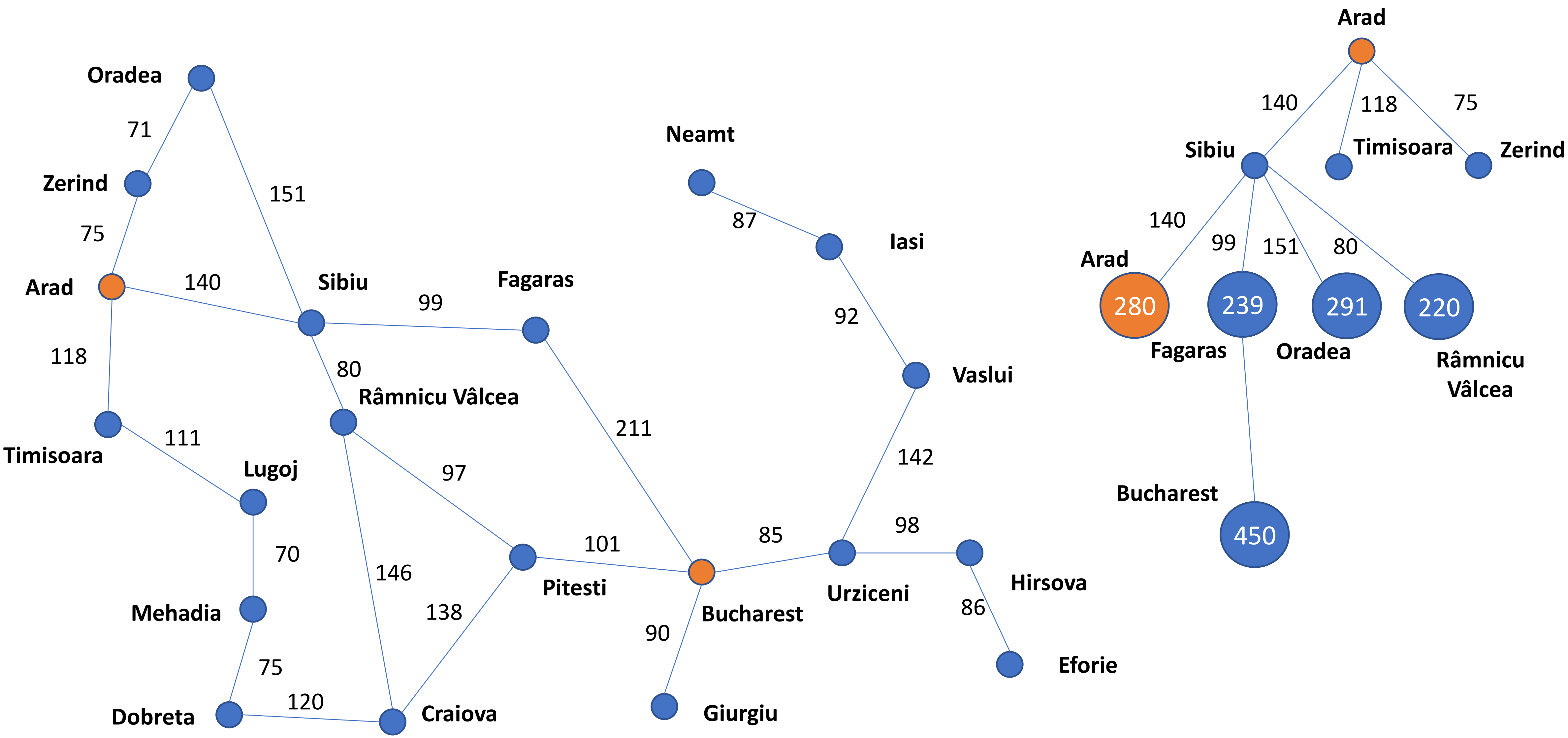


Arad
Sibiu
Timisoara
Zerind
Arad
Fagaras
Oradea
Râmnicu Vâlcea

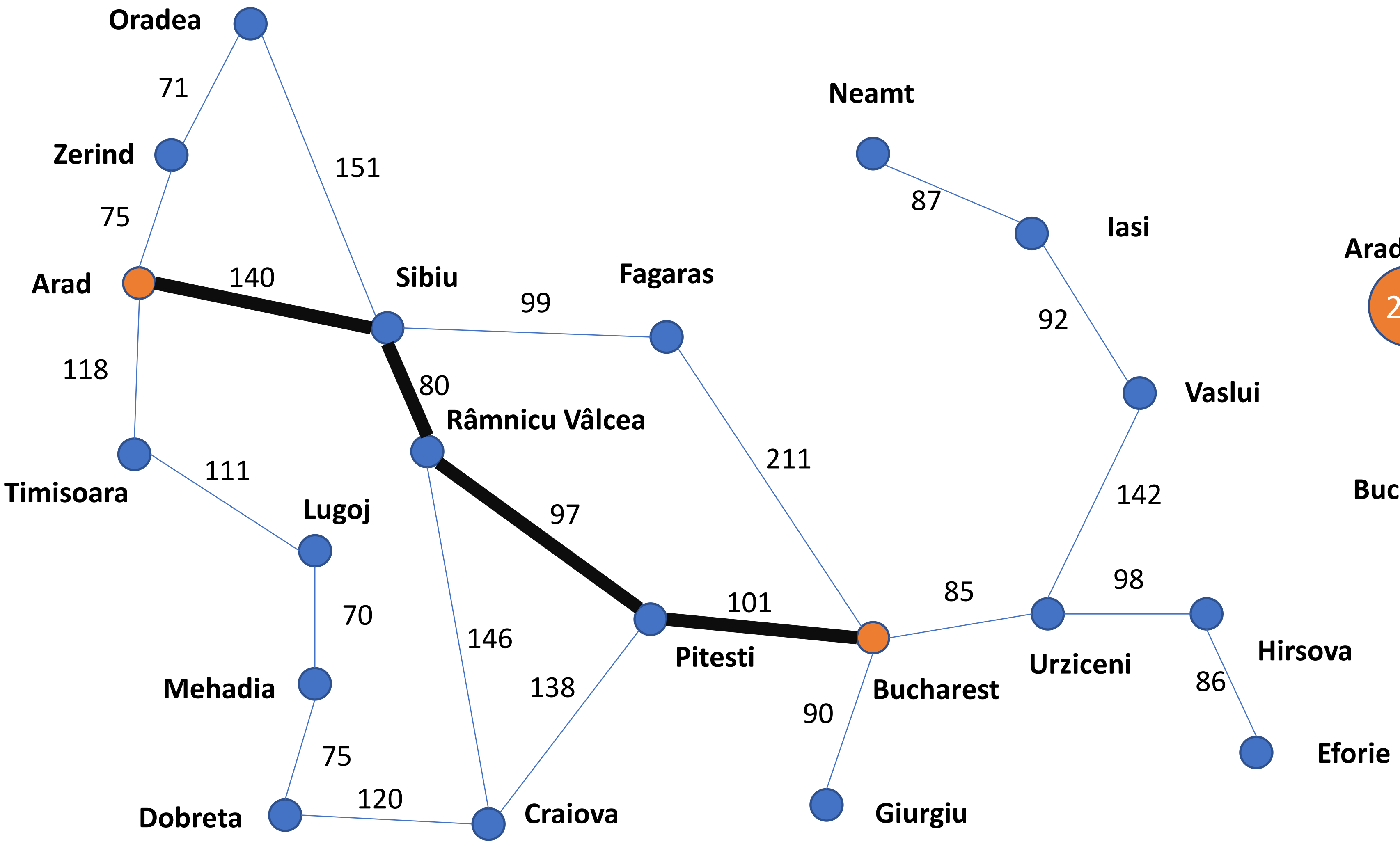


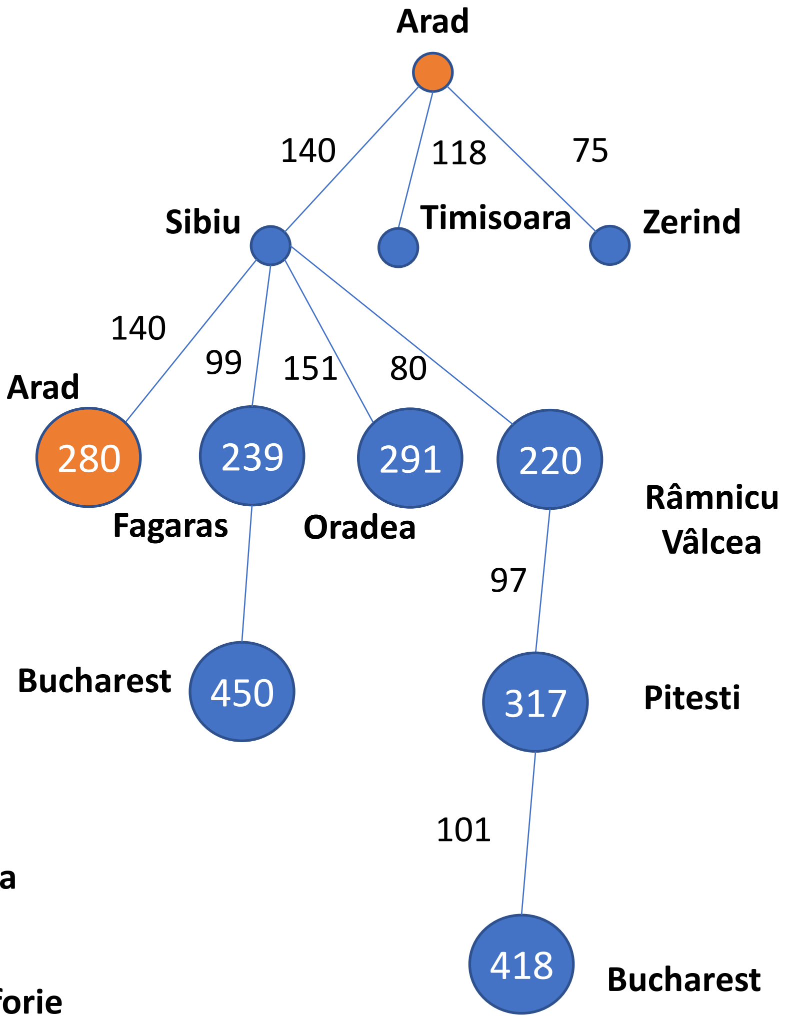
THE END

Depth First Search ~~is~~ LIES clearly the best >_>



The Best Path!





State Space and Search Nodes

Is the State Space and Search Nodes the same thing?

- No!

What is the difference?

A **state** is a representation of a physical configuration

A **node** is a data structure constituting part of a search tree

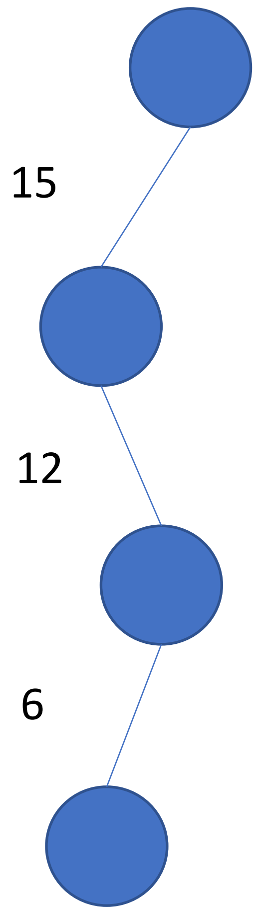
State Space and Search Nodes

Search Nodes have:

- A Parent
- Children
- Depth
- Path Cost

How many nodes deep?

Cost of nodes



Depth = 4?
(or perhaps 3?)

Path Cost =
15 + 12 + 6
= 33

How can we compare search strategies?

#1: Is it **complete**?

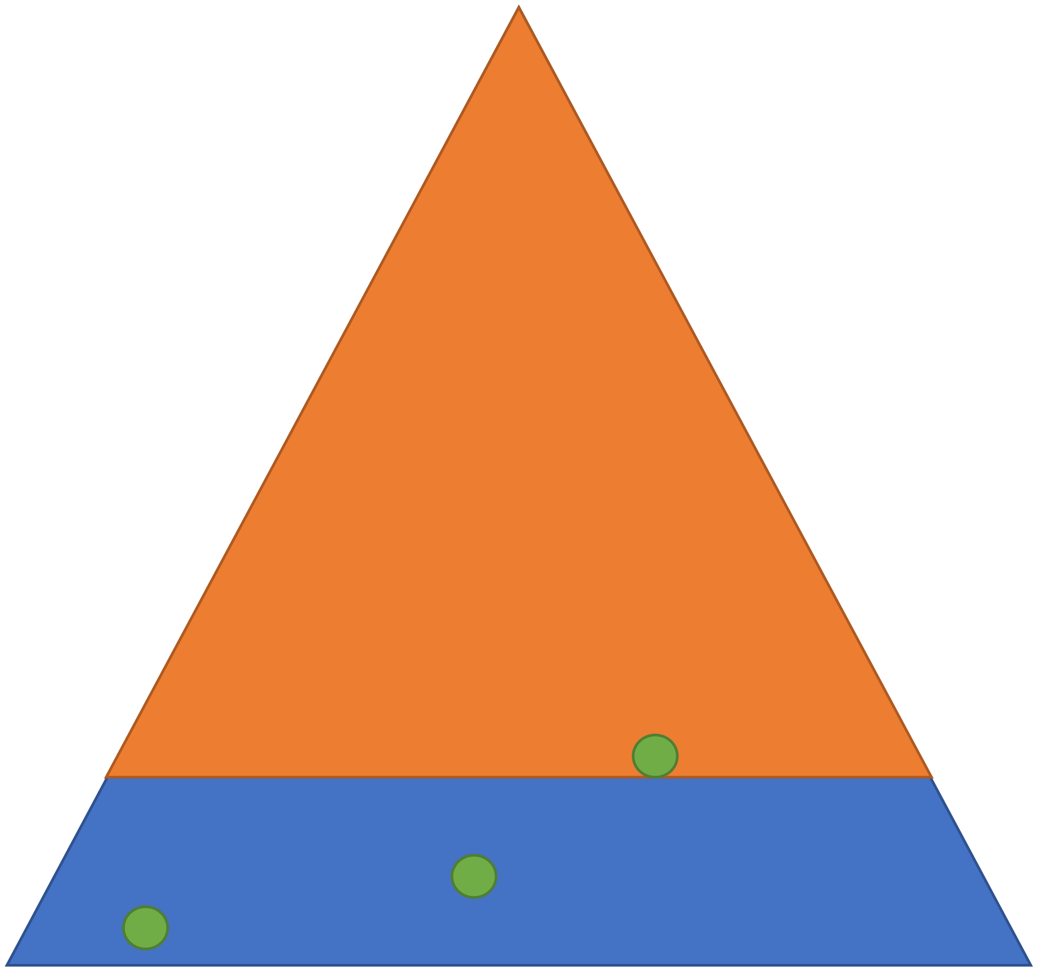
- I.e., if there is a solution, will it find the solution.

#2: Is it **optimal**?

- I.e., if it finds a solution, is that solution the best solution?

How do BFS and DFS compare?

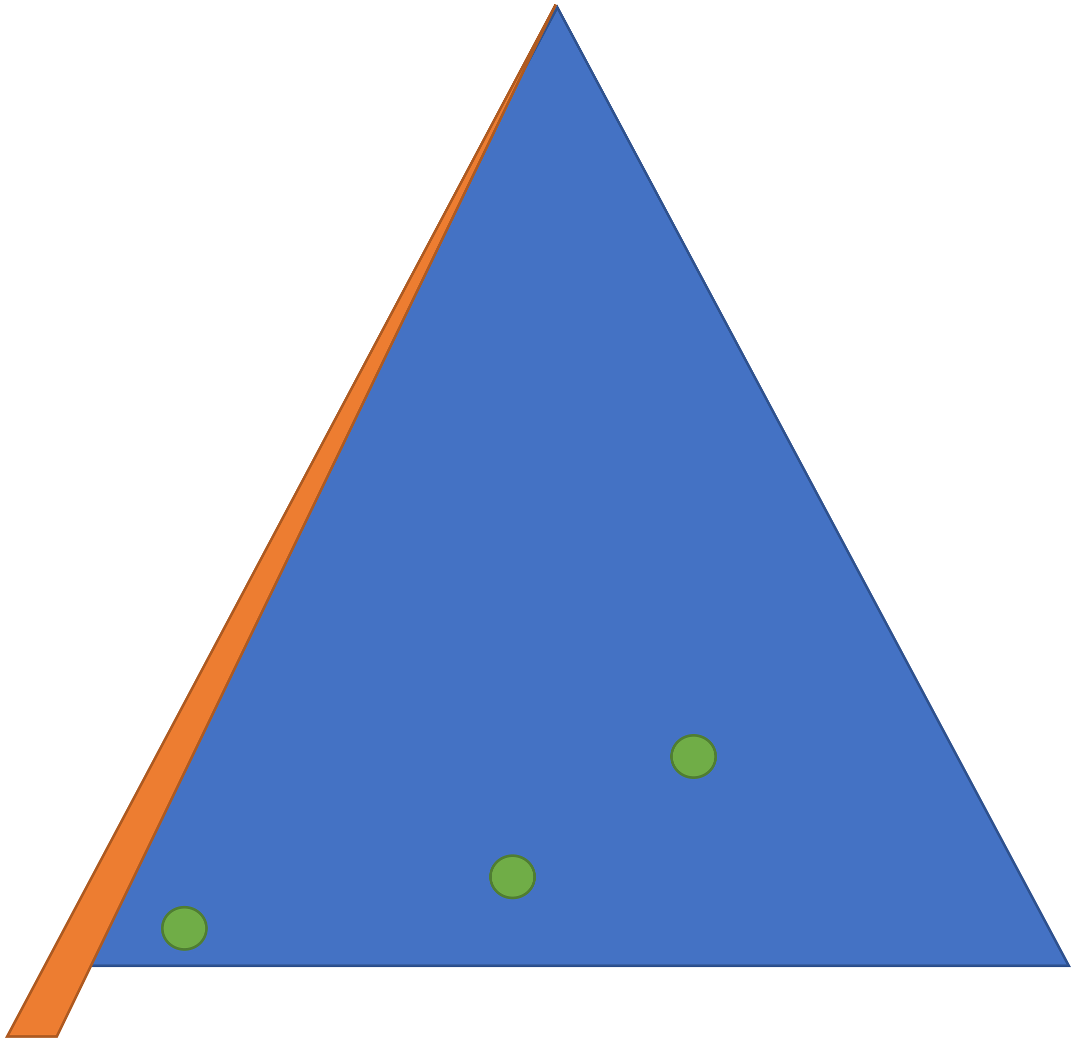
Breadth First Search



If there is a finite distance between the origin and goal…

Then BFS is complete

Depth First Search



Without marking 'nodes'

DFS is incomplete

How can we compare search strategies?

#1: Is it **complete**?

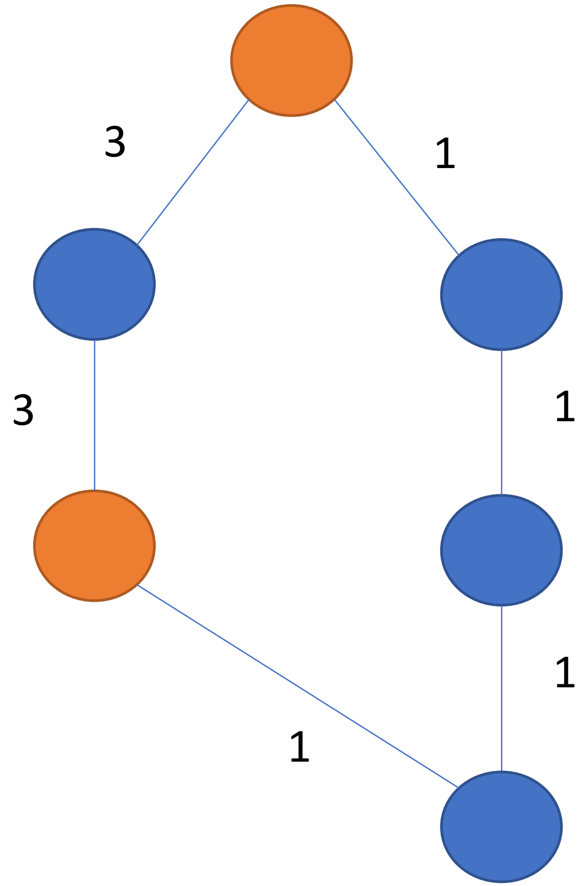
- I.e., if there is a solution, will it find the solution.

#2: Is it **optimal**?

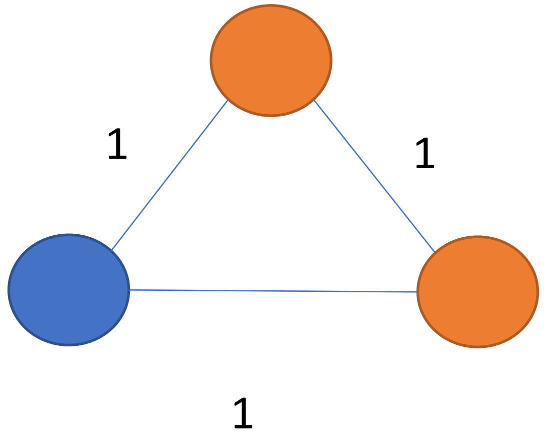
- I.e., if it finds a solution, is that solution the best solution?

	BFS	**DFS**
Complete?	Yes	Depends
Optimal?	No	No

Counterexamples…



BFS will always fail



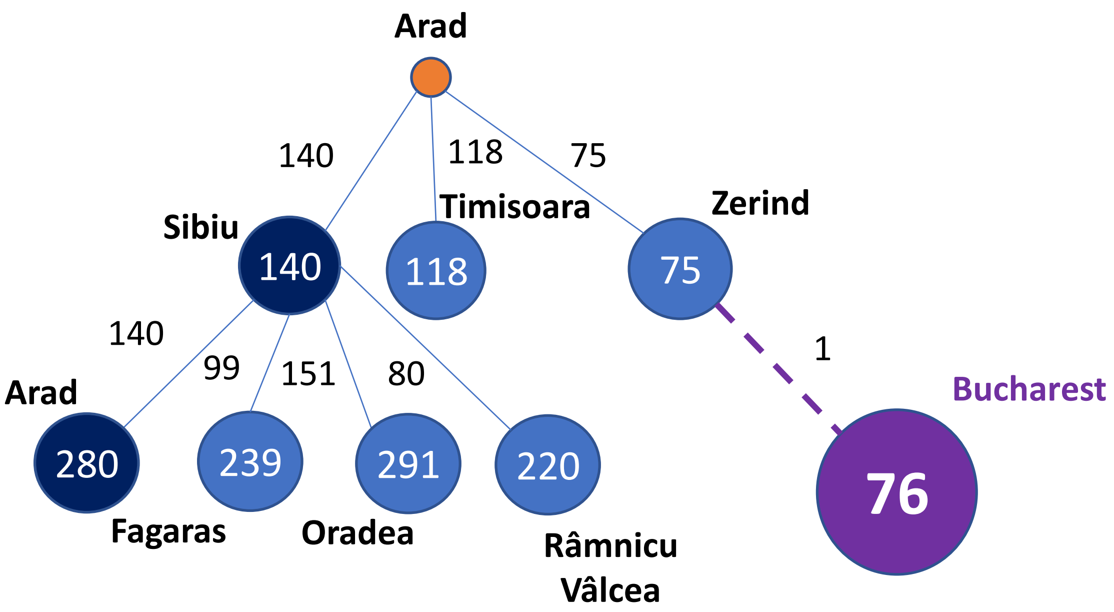
DFS will fail if it looks ‘left first’

More Optimal Searches?

Uniform Cost Search

If we stop once we find the solution, then which node must we check next?

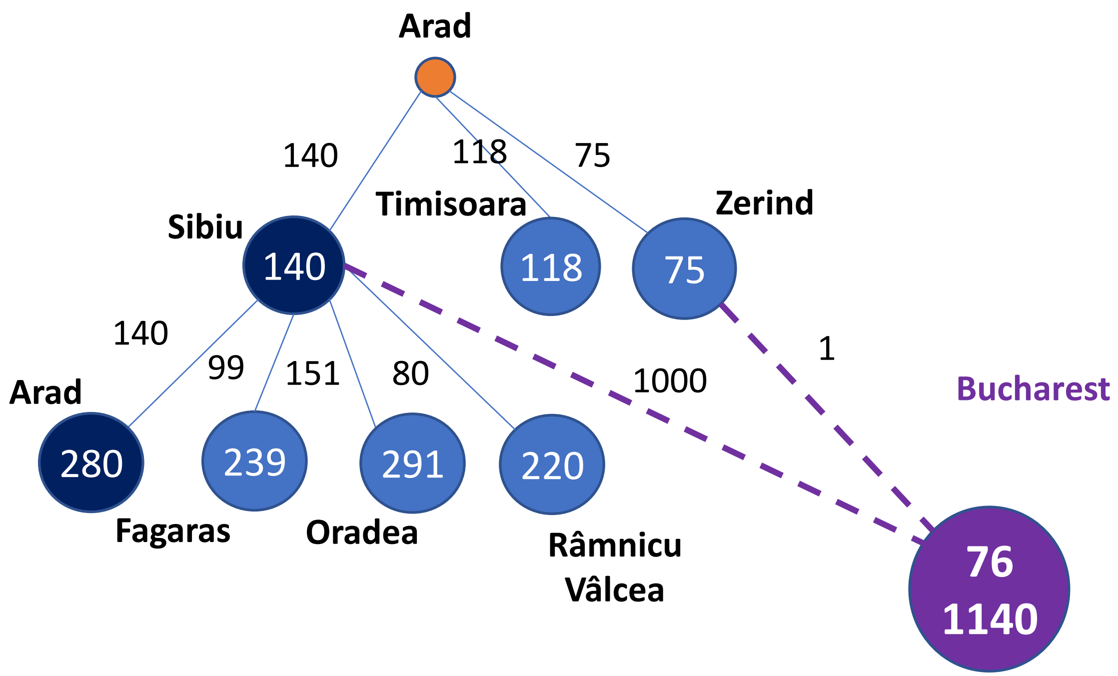
- Zerind



More Optimal Searches?

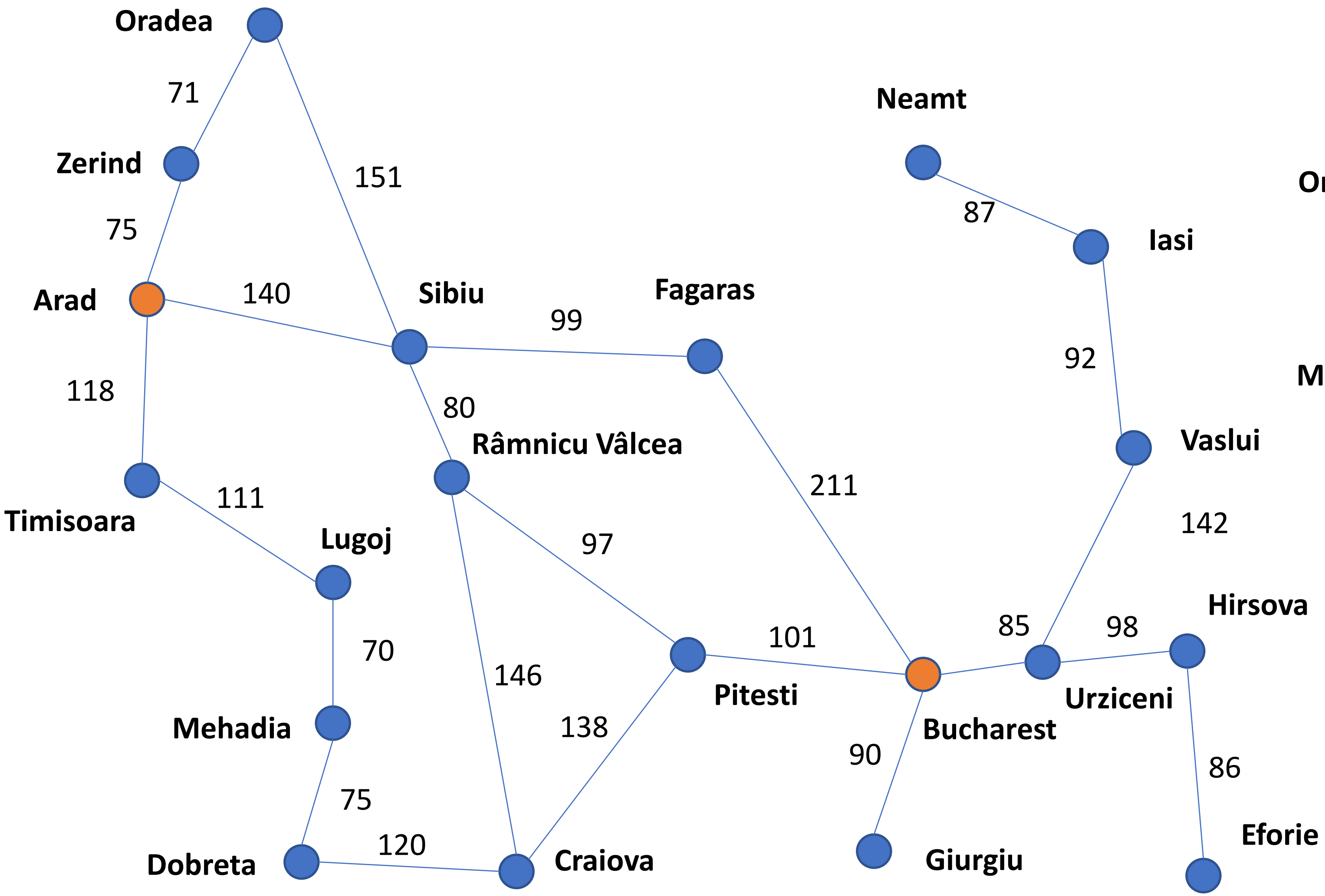
It is important we only 'count' a node when we expand it.

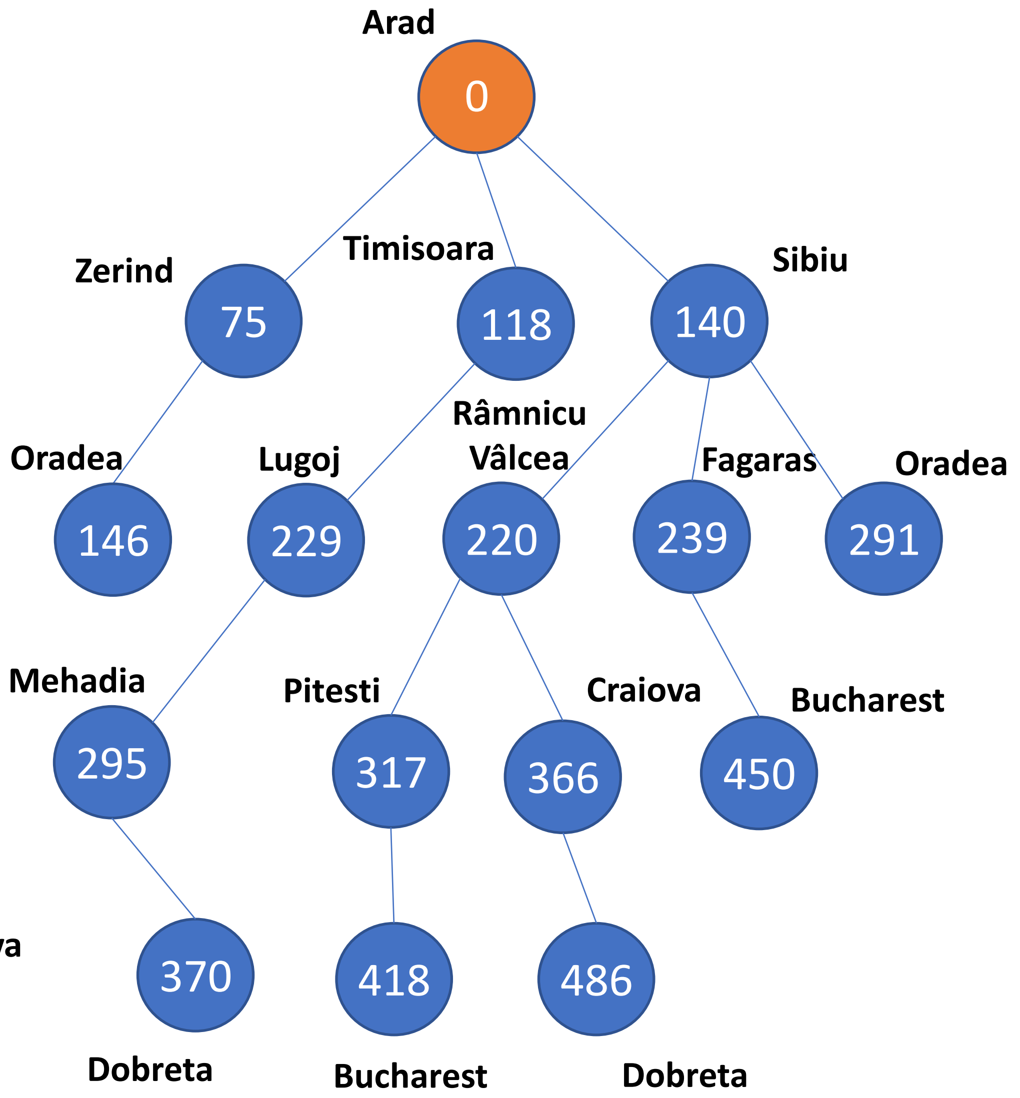
If we counted nodes as we 'added' them we could end up with this contradiction…



Of course, this is all wrong anyway… did you notice that we should never have expanded Sibiu using this logic? We should have gone for Zerind to start!!

Uniform Cost Search





Analysis

A few definitions:

- b: maximum branching factor of the search tree
- d: depth of the least cost solution
- m: maximum depth of the state space (which may be ∞)

Analysis

	BFS	DFS	UCS
Complete?	Yes	Depends	Yes
Optimal?	No	No	Yes
Space Complexity?	O(V) $O(b^{d+1})$	O(V) O(bm)	
Time Complexity?	O(V+E) $O(b^{d+1})$	O(V+E) $O(b^m)$	O((b + logV) * d)

Uninformed Search Strategies

What does uniformed mean?

- Only has access to information in the problem description…

Examples:

- Breadth first search
- Depth first search
- Uniform cost search

Some others we haven’t covered…

Search strategies:

- Depth-limited search (DLS): Like DFS but applies depth limits to avoid infinitely deep searches
- Iterative Deepening Search (IDS): Like DLS except the depth limits iteratively increase

Questions

What are your questions?