COMP9414 Tutorial

Week 2

Participation mark

- Remain in the tutorial for a reasonable amount of time
  - Total time so its okay if you disconnect a few times

- Complete the provided form
  - Sent after class, due prior to the next tutorial
  - Submit however many times you want, I'll take the latest one

News

- Assignment will be released Monday, Week 3
  - June 15th
  - Explanation during consultation time
    - Make sure you attend
- Exam will be multiple-choice

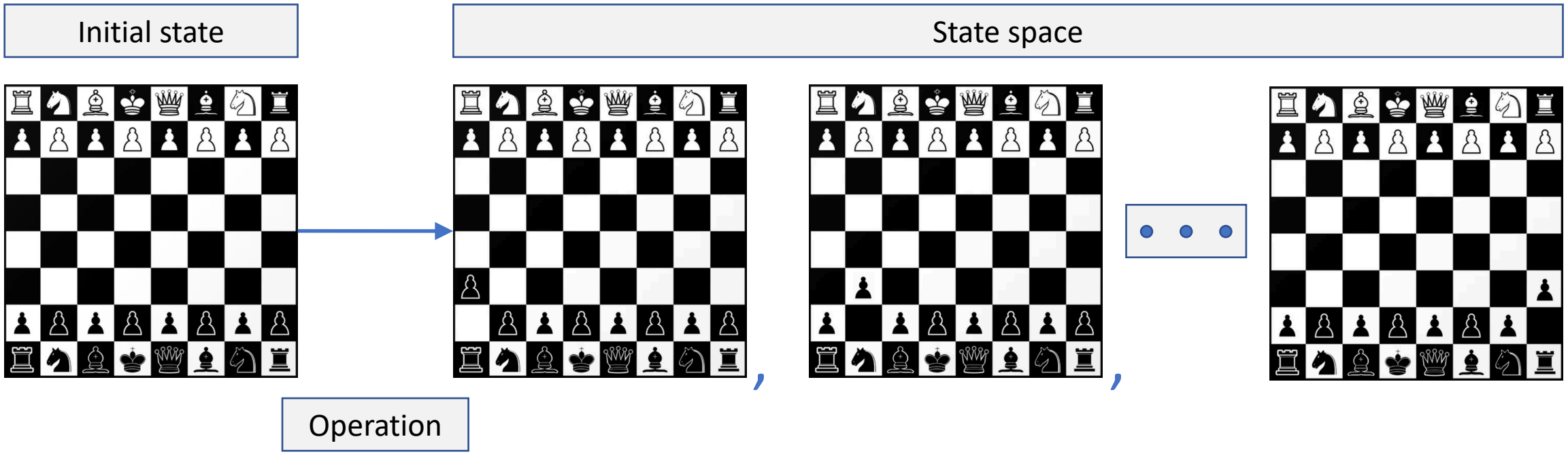
Definitions – Initial state

- Beginning state where the search algorithm would begin

- All pieces are in their initial positions
- Can start the state search from here to determine a sequence of moves to win

Definitions – State space

- Set of states reachable from the current position
  - Possible states that can be expanded from the current position
  - Potential moves that can be made



Definitions – Transitions

- Based on operations or actions that an agent can perform
- Successor function determines state space based on this operation

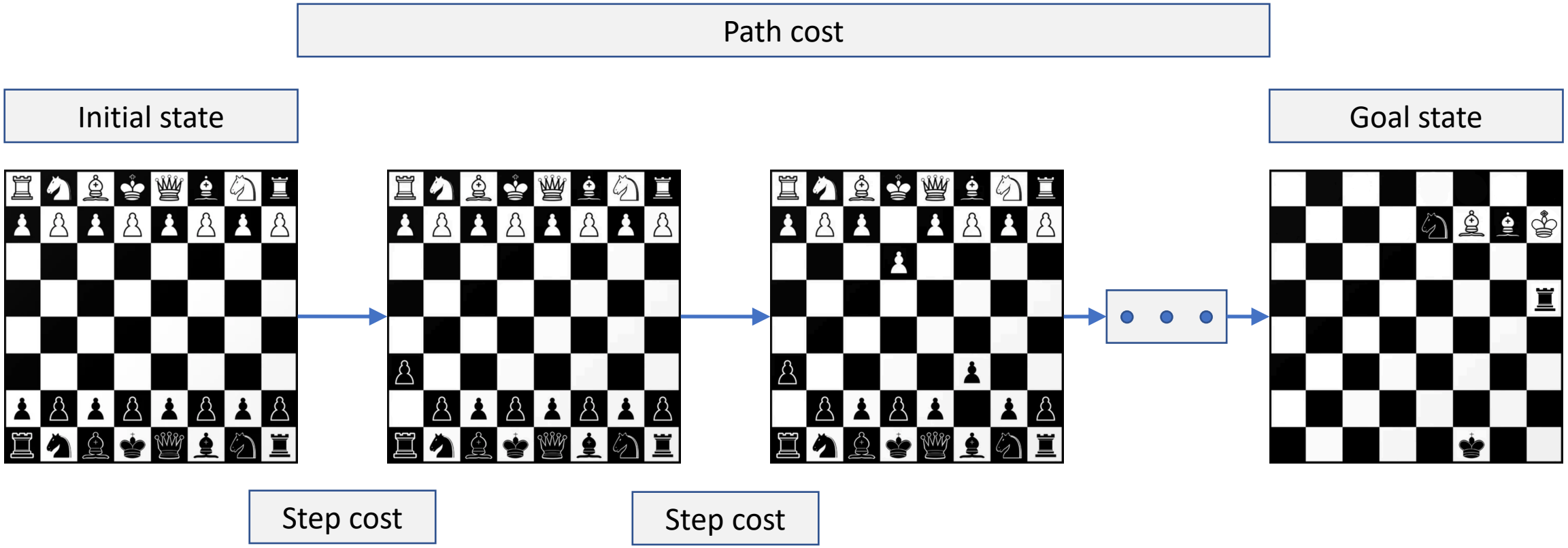
Successor Function
S (current_state, operation) = new_state

Operations	
Moving a particular piece	Achieving check
Removing an opponent piece	Achieving check-mate

S ( , Move pawn A2 → A3) =

Definitions – Traversal

- Find a path from initial state to goal state
- Path cost should be minimised



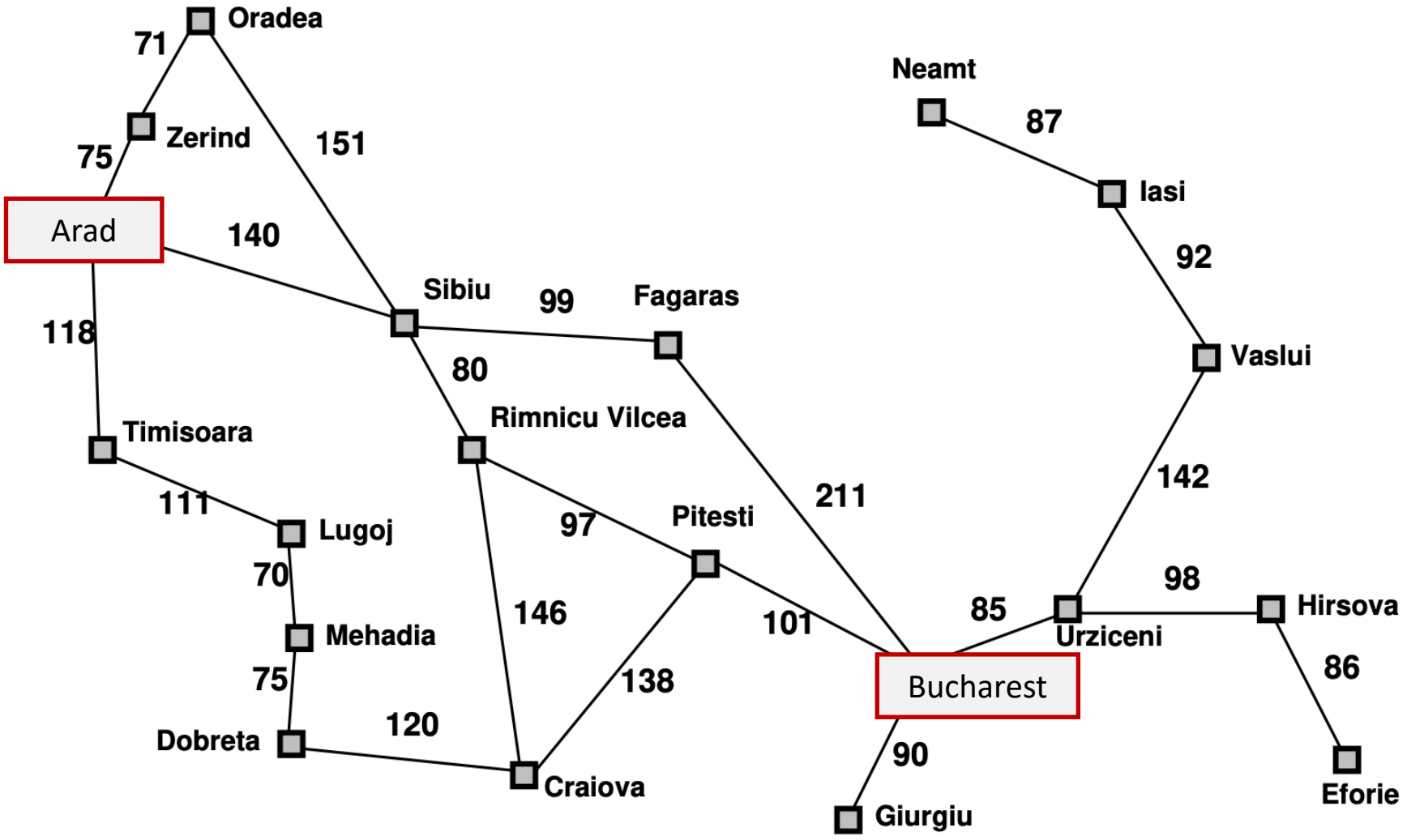
Definitions

- Initial state
  - Starting state at the beginning of the search algorithm
- State space
  - Set of states reachable from the current state
- Transition
  - Based on operations or actions that an agent can perform
  - Successor function determines state space based on this operation

Definitions

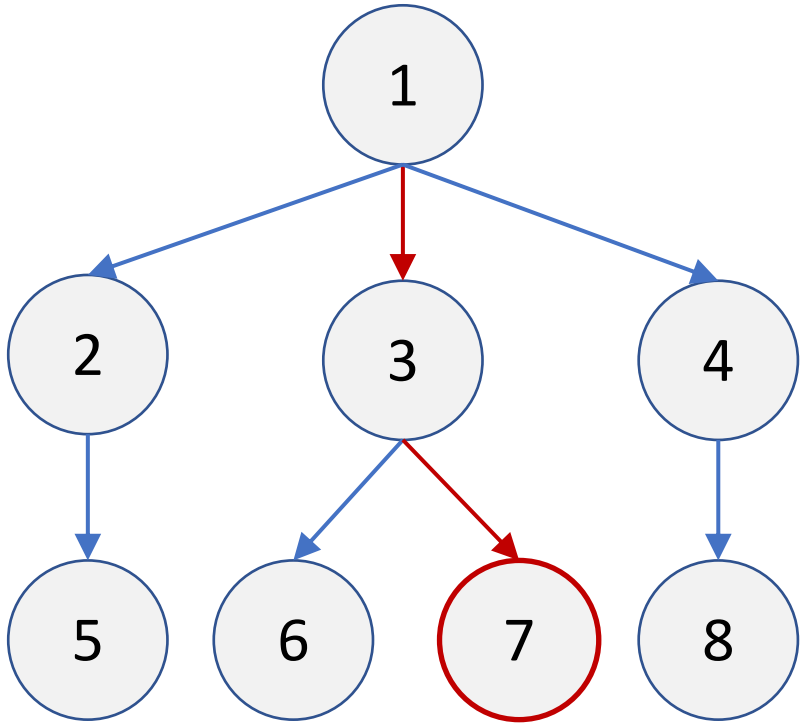
- Goal state
  - State that needs to be reached
- Step cost
  - Cost required to transition from one state to another in the immediate state space
- Path cost
  - Cost required to transition from one state to another along some path
  - Can have multiple states along the path

Question 1 – Arad → Bucharest



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

Question 1 – Depth-first Search



Time Complexity	O(bm)
Space Complexity	$O(b^m)$

Not Complete
Not Optimal

Implementation	Datatype
Stack	List
Visited list (Generation)	List
Connections	Dictionary

Stack											
1	2	3	6	7							

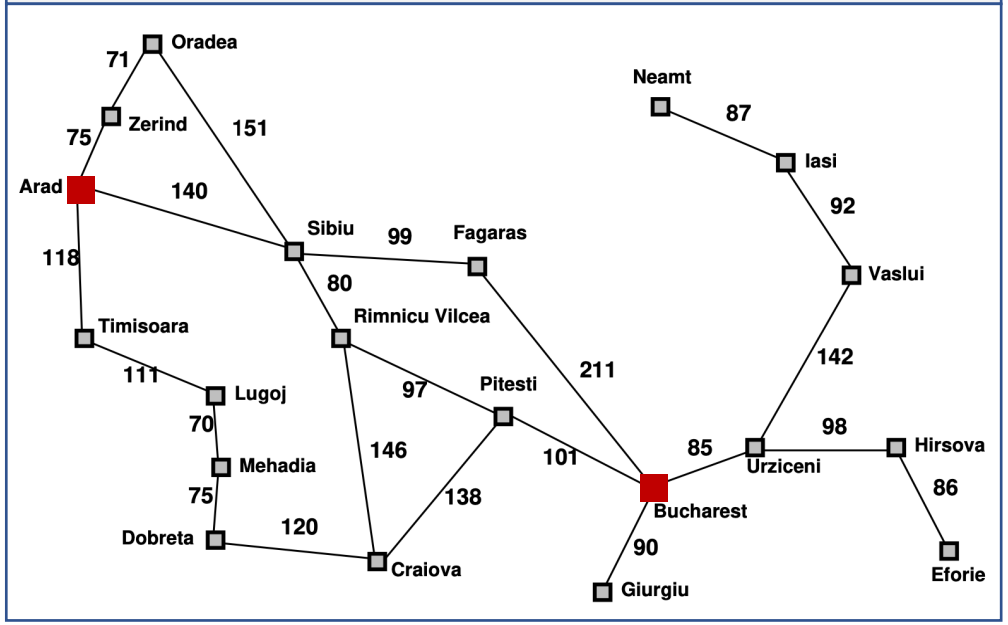
Question 1 – Depth-first Search

Visited list		
A	Z	T
S	R	F
B		

Connections			
/ : A	A : Z	A : T	A : S
S : R	S : F	F : B	:
:	:	:	:

Selected Path			
A	S	F	B

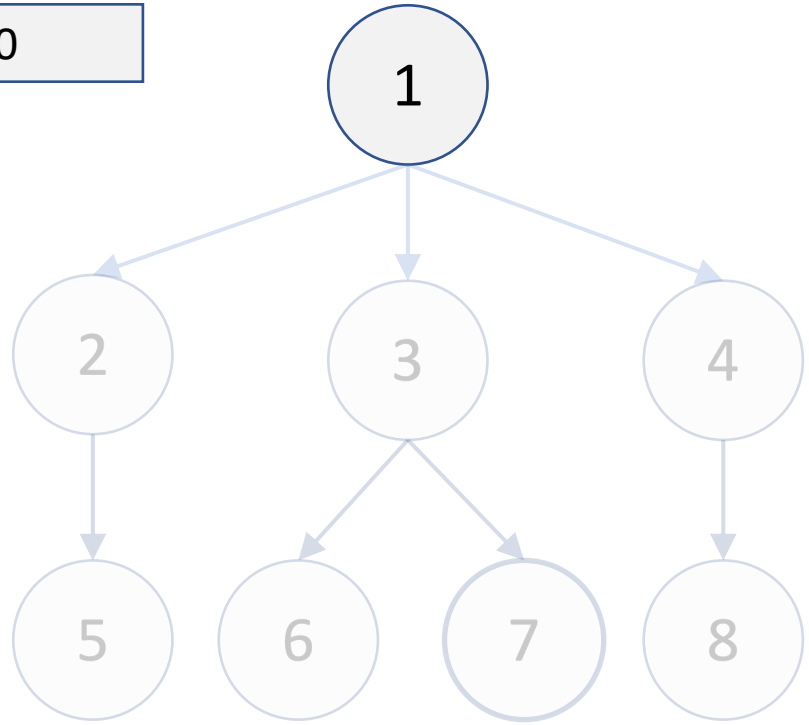
Alphabetical order



Stack																				
A	Z	T	S	R	F	B														

Question 1 – Iterative-Deepening Search

Depth = 0



Time Complexity	$O(b^d)$
Space Complexity	O(d)

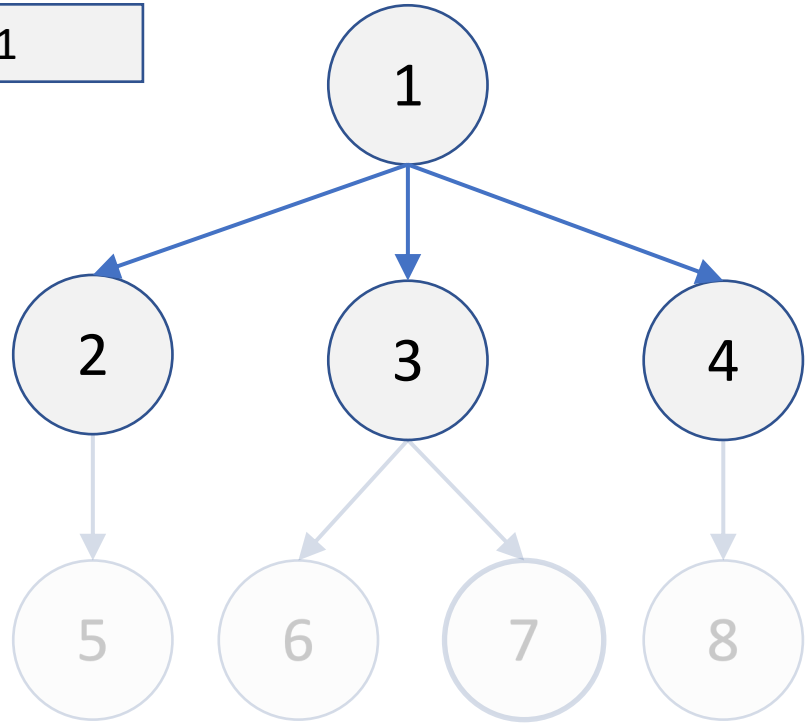
Complete
Optimal

Implementation	Datatype
Stack	List
Visited list (Generation)	List
Connections	Dictionary

Stack											

Question 1 – Iterative-Deepening Search

Depth = 1



Time Complexity	$O(b^d)$
Space Complexity	O(d)

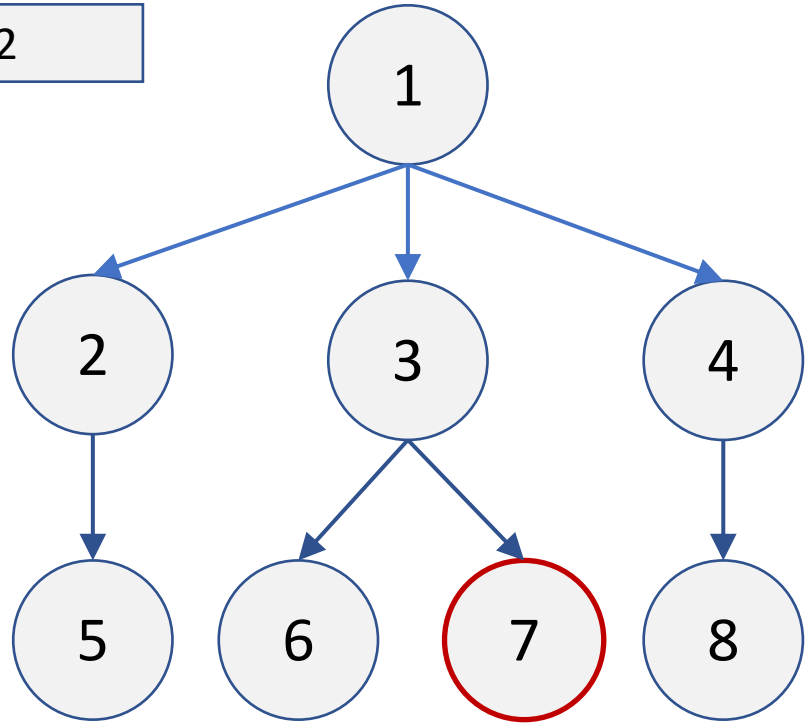
Complete
Optimal

Implementation	Datatype
Stack	List
Visited list (Generation)	List
Connections	Dictionary

Stack											

Question 1 – Iterative-Deepening Search

Depth = 2



Time Complexity	$O(b^d)$
Space Complexity	O(d)

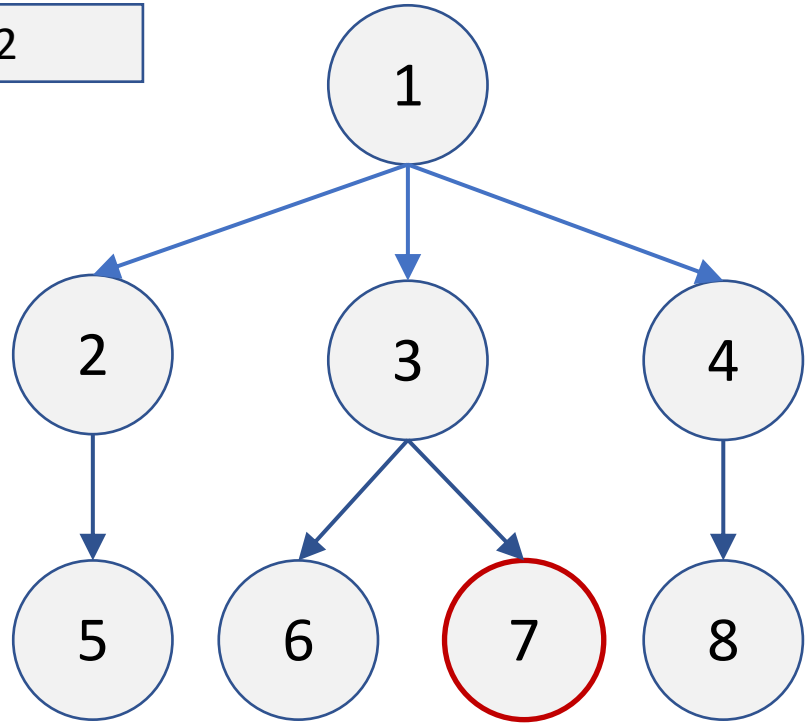
Complete
Optimal

Implementation	Datatype
Stack	List
Visited list (Generation)	List
Connections	Dictionary

Stack											

Question 1 – Iterative-Deepening Search

Depth = 2



Time Complexity	$O(b^d)$
Space Complexity	O(d)

Complete
Optimal

Implementation	Datatype
Stack	List
Visited list (Generation)	List
Connections	Dictionary

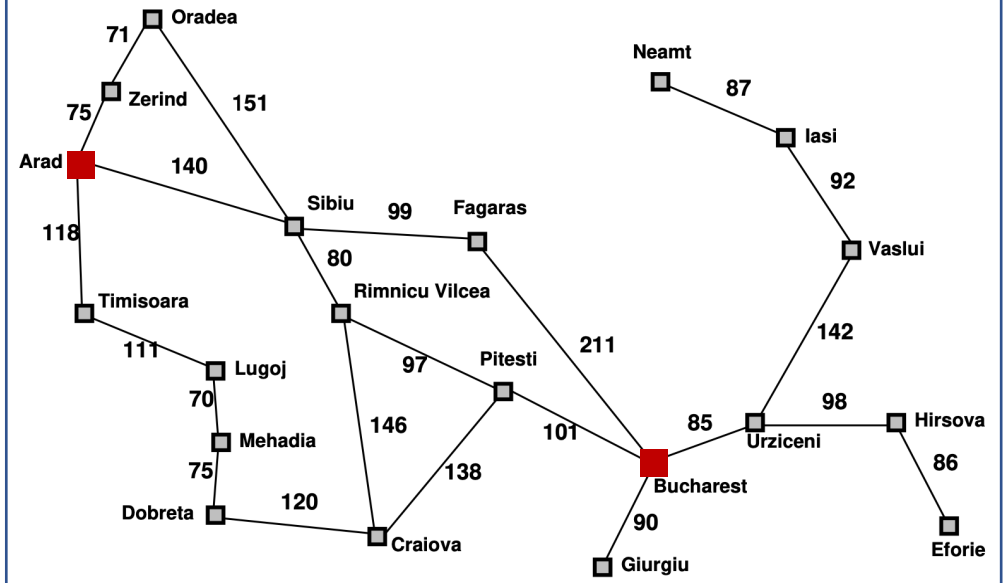
Stack											
1	2	3	6	7							

Question 1 – Iterative-Deepening Search

Depth = 0		
Visited list		
A		

Connections			
/ : A	:	:	:
:	:	:	:
:	:	:	:

Alphabetical order



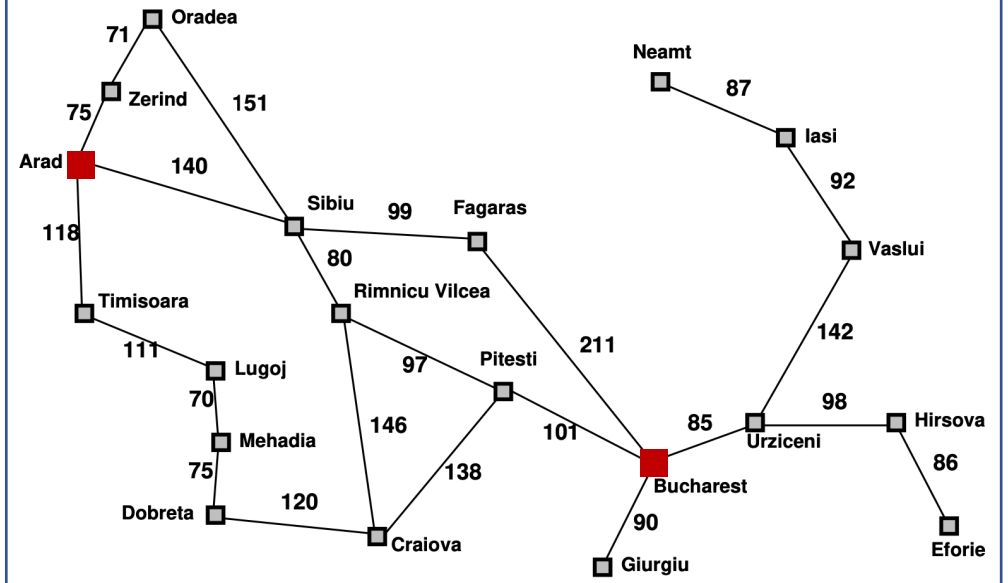
Stack																			

Question 1 – Iterative-Deepening Search

Depth = 1		
Visited list		
A	S	T
Z		

Connections			
/ : A	A : Z	A : T	A : S
:	:	:	:
:	:	:	:

Alphabetical order



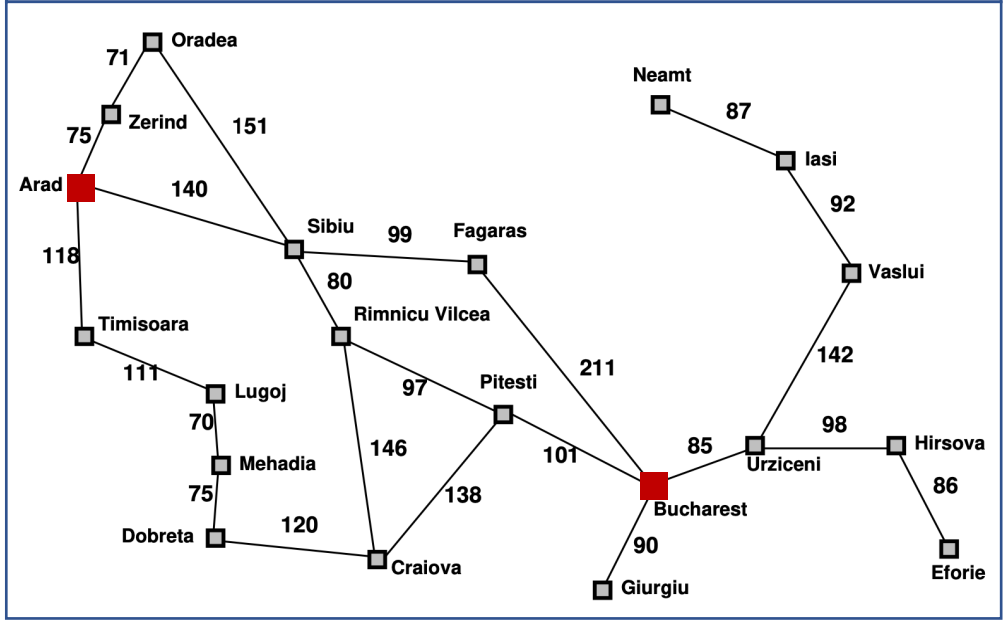
Stack																			

Question 1 – Iterative-Deepening Search

Depth = 2		
Visited list		
A	Z	T
S	O	R
F	L	

Connections			
/ : A	A : Z	A : T	A : S
S : O	S : R	S : F	T : L
:	:	:	:

Alphabetical order



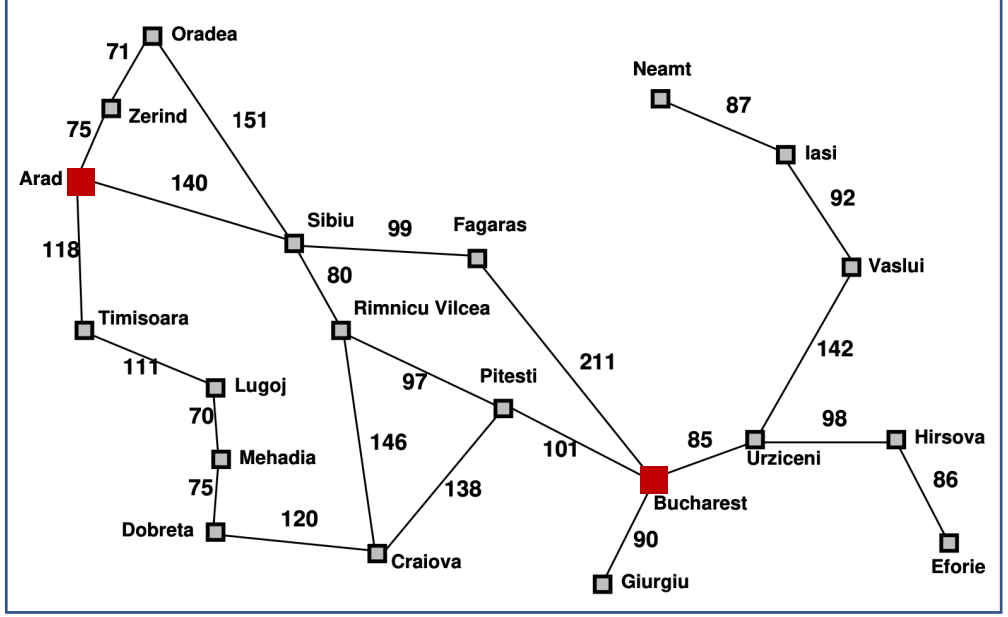
Stack																			

Question 1 – Iterative-Deepening Search

Depth = 3		
Visited list		
A	Z	T
S	O	R
F	B	

Connections			
/ : A	A : Z	A : T	A : S
S : O	S : R	S : F	S : B
:	:	:	:

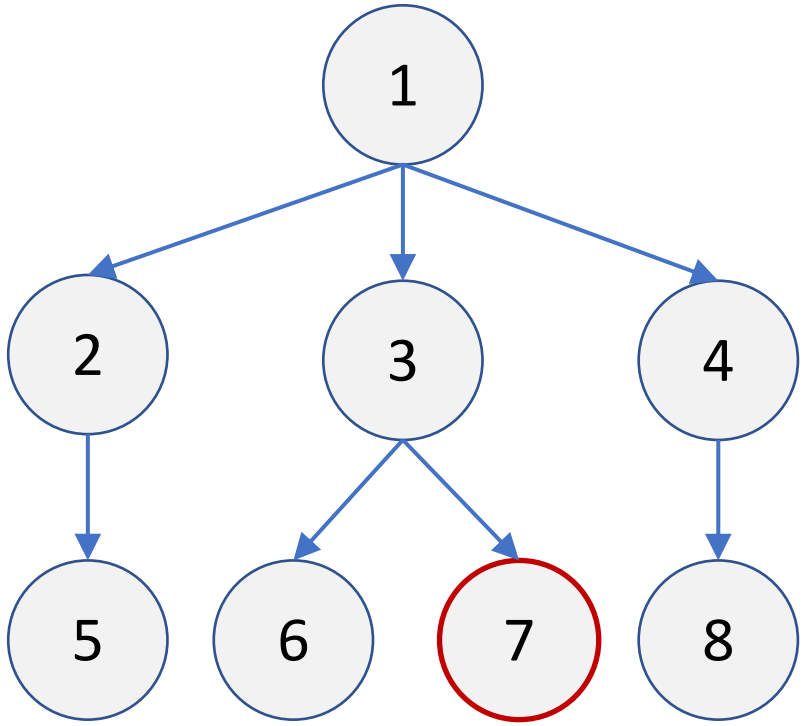
Alphabetical order



Selected Path			
A	S	F	B

Stack																			
A	Z	T	S	O	R	F	B												

Question 1 – Breadth-first Search



Time Complexity	$O(b^d)$
Space Complexity	$O(b^d)$

Complete
Optimal

Implementation	Datatype
Queue	List
Visited list (Generation)	List
Connections	Dictionary

Queue											
		3	4	5	6	7					

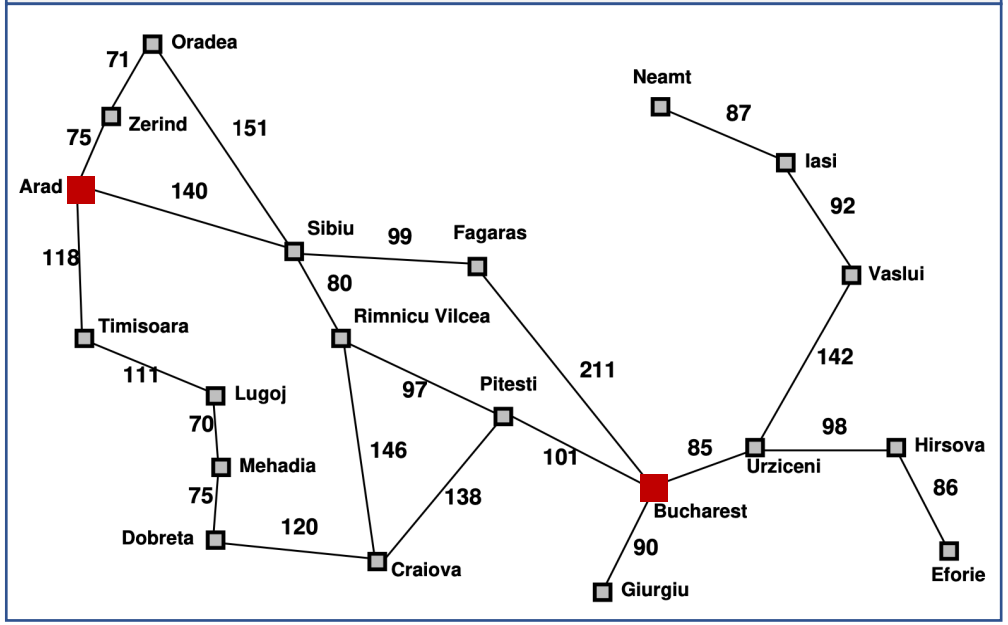
Question 1 – Breadth-first Search

Visited list		
A	S	T
Z	F	O
R		

Connections			
/ : A	A : S	A : T	A : Z
S : F	S : O	S : R	T : L
F : B	:	:	:

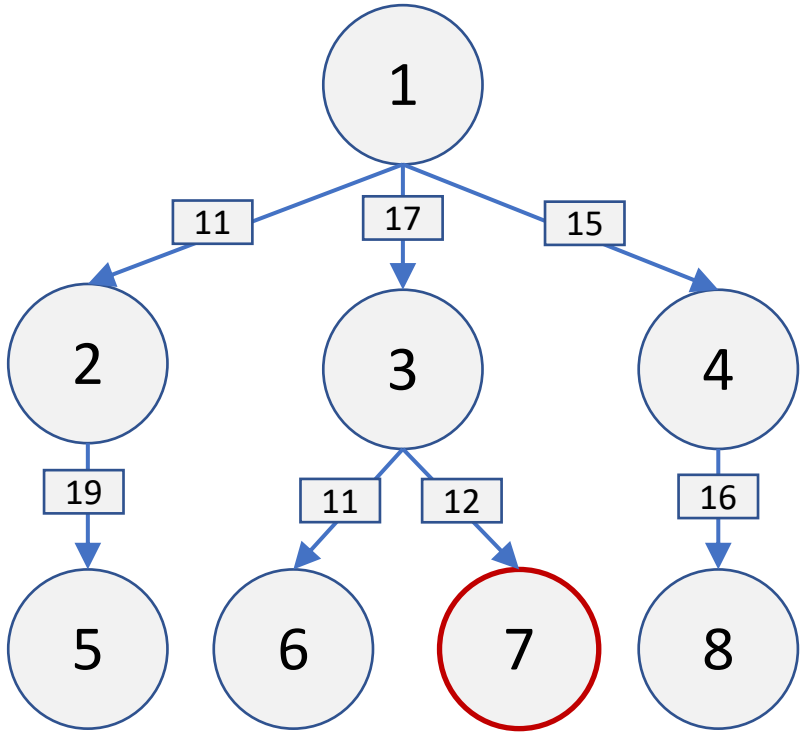
Selected Path			
A	S	F	B

Alphabetical order



Queue																			
				F	O	R	L	B											

Question 1 – Uniform-cost Search



C* = Optimal path cost	e = minimum step cost
Time Complexity	$O(b^{C^*/e})$
Space Complexity	$O(b^{C^*/e})$

Complete	Ordered by path cost
Optimal	Find goal state on expansion

Implementation	Datatype
Priority Queue	List
Visited list (Expansion)	List
Connections	Dictionary

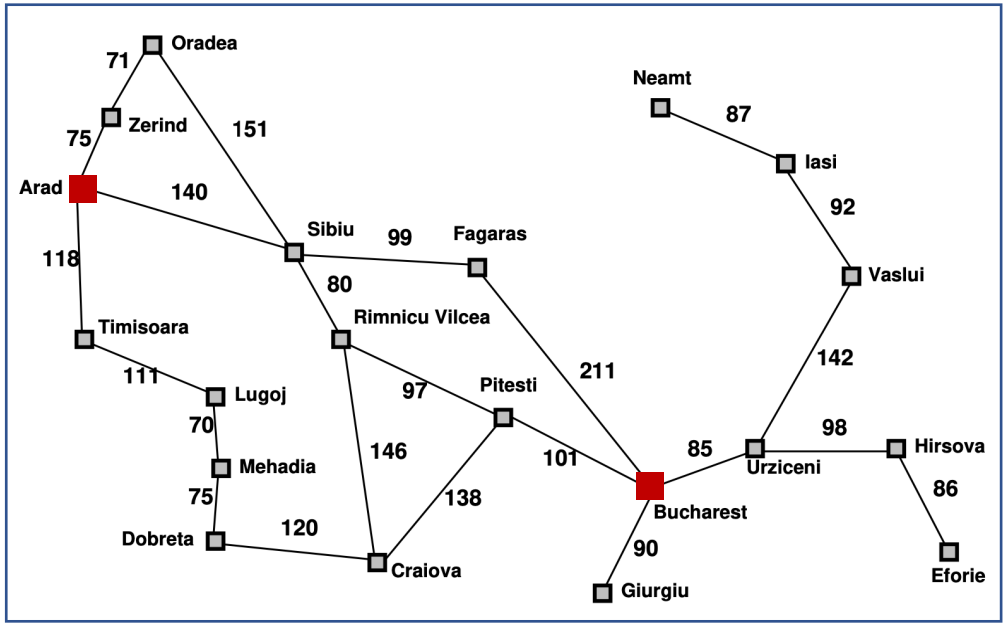
Priority Queue											
			29	30	31						
			7	5	8						

Question 1 – Uniform-cost Search

Visited list		
A	Z	T
S	O	R
L	F	M
P	C	D
B		

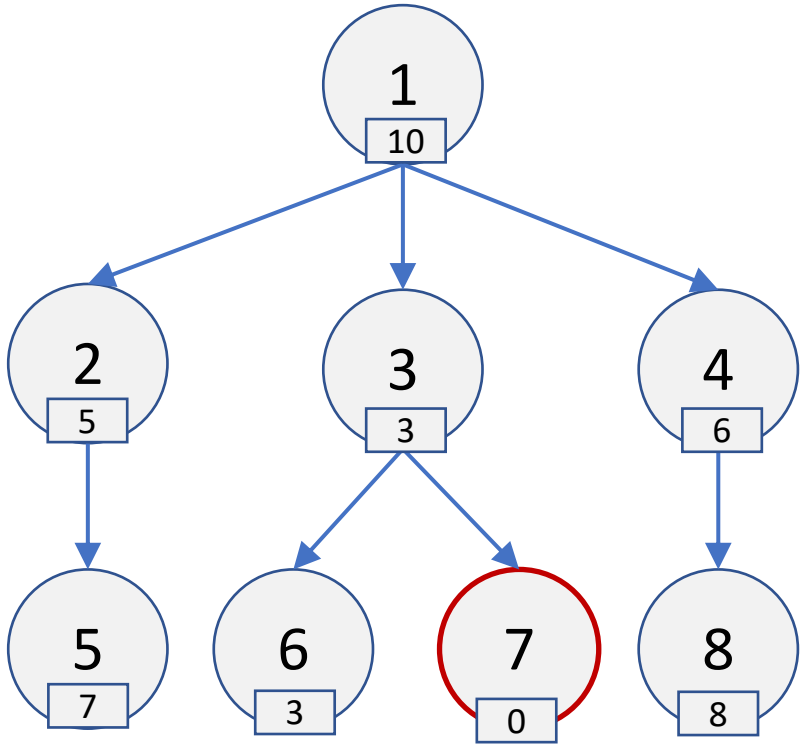
Connections			
/ : A	A : Z	A : T	A : S
Z : O	T : L	S : F	S : R
R : C	R: P	L : M	P : B
M : D	:	:	:

Selected Path				
A	S	R	P	B



Priority Queue																				
											418									
											B									

Question 1 – Greedy Best-first Search



Time Complexity	$O(b^m)$
Space Complexity	$O(b^m)$

Ordered by heuristic	Not Complete
	Not Optimal

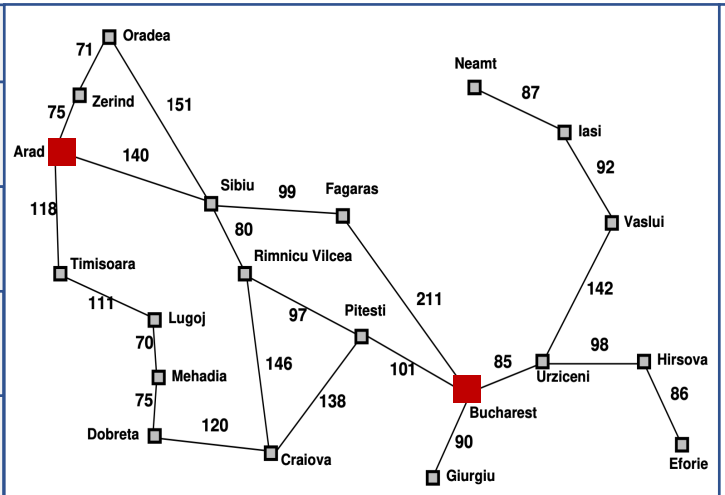
Implementation	Datatype
Priority Queue	List
Visited list (Expansion)	List
Connections	Dictionary

Priority Queue											
0	3	5	6								
7	6	2	4								

Question 1 – Greedy Best-first Search

Visited list		
A	S	T
Z	F	R
O	B	

Connections			
/ : A	A : S	A : T	A : Z
S : F	S : R	S : O	F : B
:	:	:	:
:	:	:	:

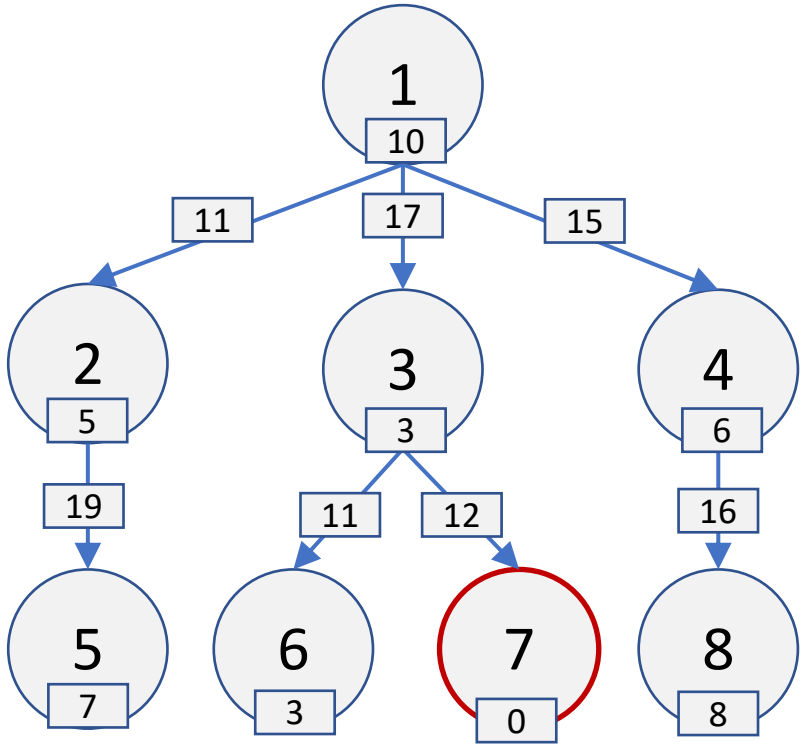


Straight–line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Selected Path			
A	S	F	B

Priority Queue															
							0	329	374	380					
							B	T	Z	O					

Question 1 – A* Search



Time Complexity	$O(b^d)$
Space Complexity	$O(b^d)$

Complete	Ordered by path cost + heuristic
Optimal	Find goal state on expansion

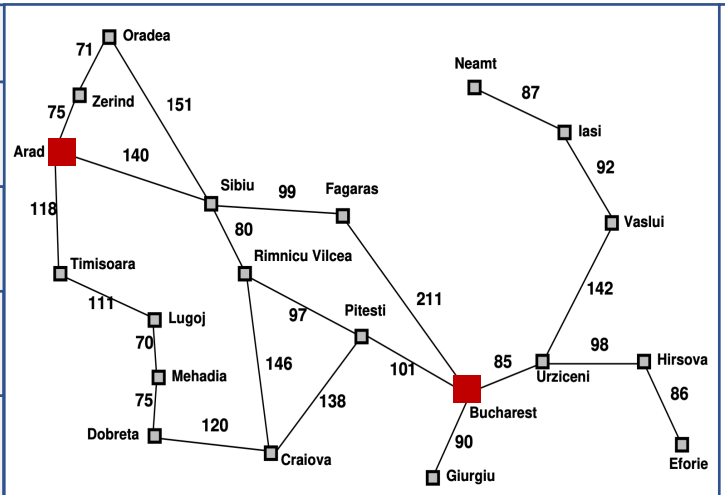
Implementation	Datatype
Priority Queue	List
Visited list (Expansion)	List
Connections	Dictionary

Priority Queue											
				29	31	37	39				
				7	6	5	8				

Question 1 – A* Search

Visited list		
A	S	R
P	F	B

Connections			
/ : A	A : S	A : T	A : Z
S : R	S : F	S : O	R : P
R : C	P : B	:	:
:	:	:	:



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

Selected Path				
A	S	R	P	B

Priority Queue																	
								418	447	449	526	671					
								B	T	Z	C	O					

A* Search Optimality

- Optimal and complete depending on the heuristic
- Heuristic must be an underestimate
  - Must be smaller or equal to the remaining path cost
- $h(x) \leq c(x, y) + h(y)$
  - Heuristic value of the current node x must be smaller than the step cost plus the heuristic value of the next node y

Notes

- All python files for question 3 can be found at:
  - https://artint.info/AIPython/