UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Workbook:
Breadth-first search

Albert Sanchis
Jorge Civera

DSIC

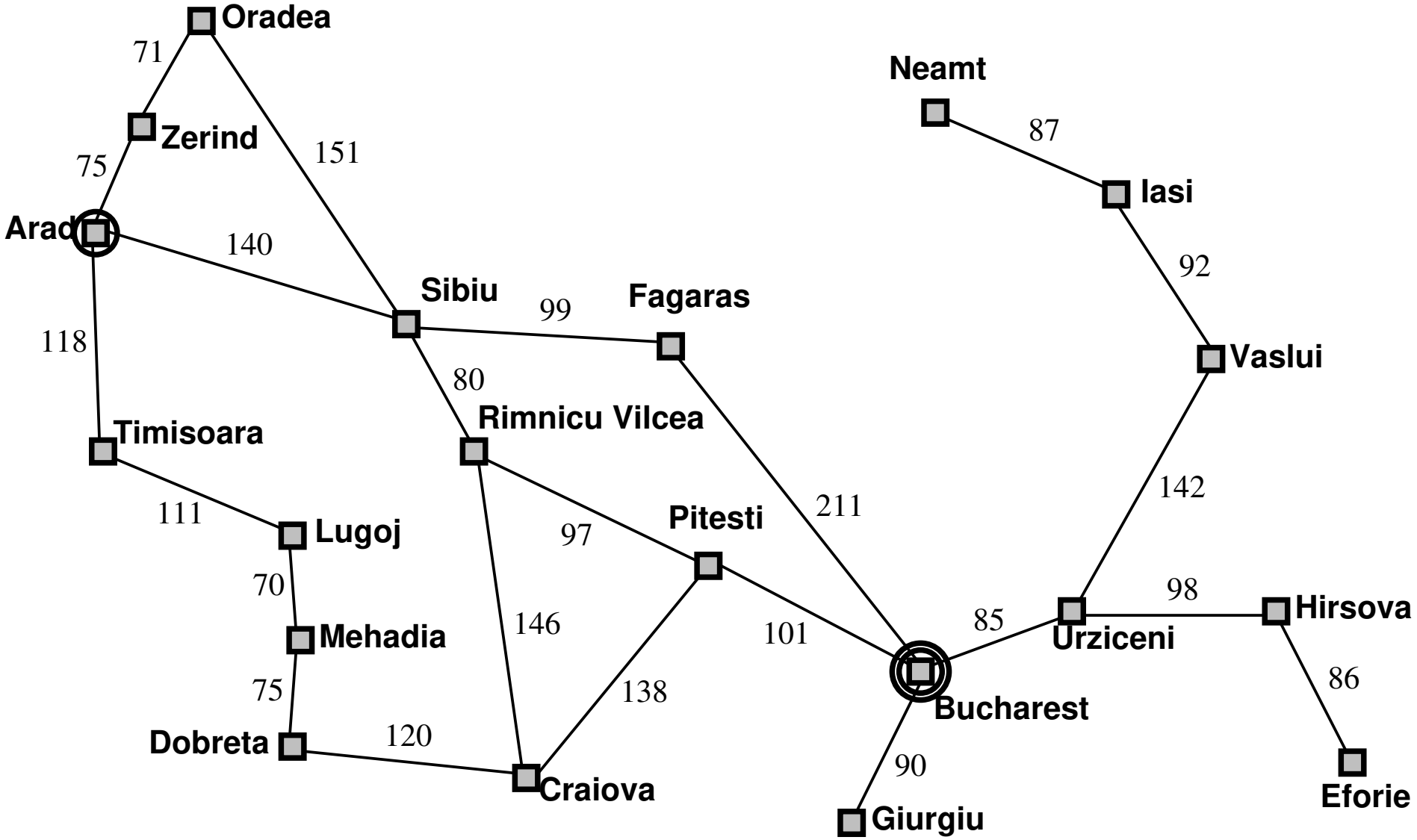
Departament de Sistemes Informàtics i Computació

Learning objectives

- To describe breadth-first search.
- To draw the tree of breadth-first search.
- To apply breadth-first search to a well-known problem.
- To analyze the quality of breadth-first search.

Problem: Shortest path between two points

Shortest path from Arad to Bucarest [1]:



Actions(Arad) = {Move(Sibiu), Move(Timisoara), Move(Zerind)}.

Breadth-first search [1, 2, 3, 4]

BFS(G, s') // ***Breadth-first search***; G graph and s' initial node

$O = InitQueue(s')$ // ***Open: search frontier-queue***

$C = \emptyset$ // ***Closed:*** set of explored nodes

while not $EmptyQueue(O)$:

$s = Unqueue(O)$ // ***FIFO (First in, first out) selection***

$C = C \cup \{s\}$ // s is now explored

forall $(s, n) \in Adjacents(G, s)$: // generation: n child of s

if $n \notin C \cup O$: // n not discovered until now

if $Goal(n)$ **return** n // solution found!

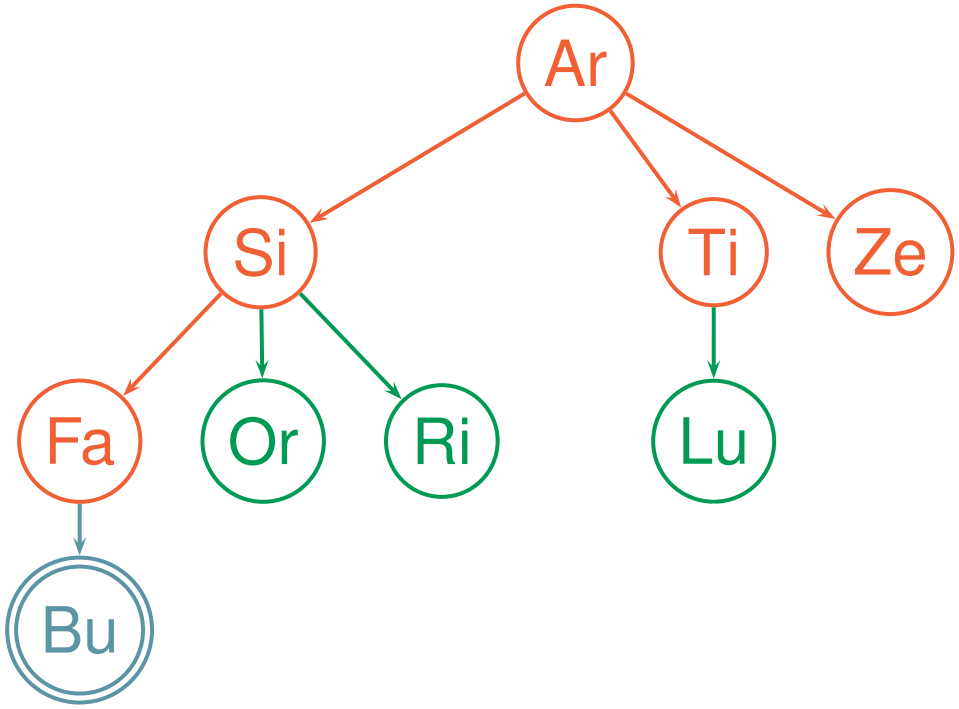
$Append(O, n)$ // n added to the queue

return NULL // no solution found

- ***Question 1***: Write a trace of the ***BFS*** algorithm applied to the problem of finding the shortest path from Arad to Bucarest.

O	C	s
{Arad}	{}	–
{Sibiu, Timisoara, Zerind}	{Arad}	Arad
{Timisoara, Zerind, Fagaras, Oradea, Rimnicu}	{Arad, Sibiu}	Sibiu
{Zerind, Fagaras, Oradea, Rimnicu, Lugoj}	{Arad, Sibiu, Timisoara}	Timisoara
{Fagaras, Oradea, Rimnicu, Lugoj}	{Arad, Sibiu, Timisoara, Zerind}	Zerind
{Oradea, Rimnicu, Lugoj}	{Arad, Sibiu, Timisoara, Zerind, Fagaras}	Fagaras

- ***Question 2***: Draw the search tree as a result of applying the ***BFS*** algorithm to the problem of finding the shortest path from Arad to Bucarest.



- ***Question 3***: Does the BFS algorithm find a solution? ***Yes***

- ***Question 4***: If the answer is "Yes":
  - What is the solution found? ***The solution path is: Arad, Sibiu, Fagaras, Bucharest***
  - What is the cost of this solution? ***450***
  - Is this the solution of minimum cost? ***No, because there is an alternative solution with lower cost of 418: Arad, Sibiu, Rimnicu, Pitesti, Bucharest***
  - What type of solution is found by the BFS algorithm? ***The shortest path in terms of number of movements***

BFS (check goal when expanding) [1, 2, 3, 4]

BFS(G, s') // ***Breadth-first search***; G graph and s' initial node

$O = InitQueue(s')$ // ***Open: search frontier-queue***

$C = \emptyset$ // ***Closed:*** set of explored nodes

while not $EmptyQueue(O)$:

$s = Unqueue(O)$ // ***FIFO (First in, first out) selection***

if $Goal(s)$ **return** s // solution found!

$C = C \cup \{s\}$ // s is now explored

forall $(s, n) \in Adjacents(G, s)$: // generation: n child of s

if $n \notin C \cup O$: // n not discovered until now

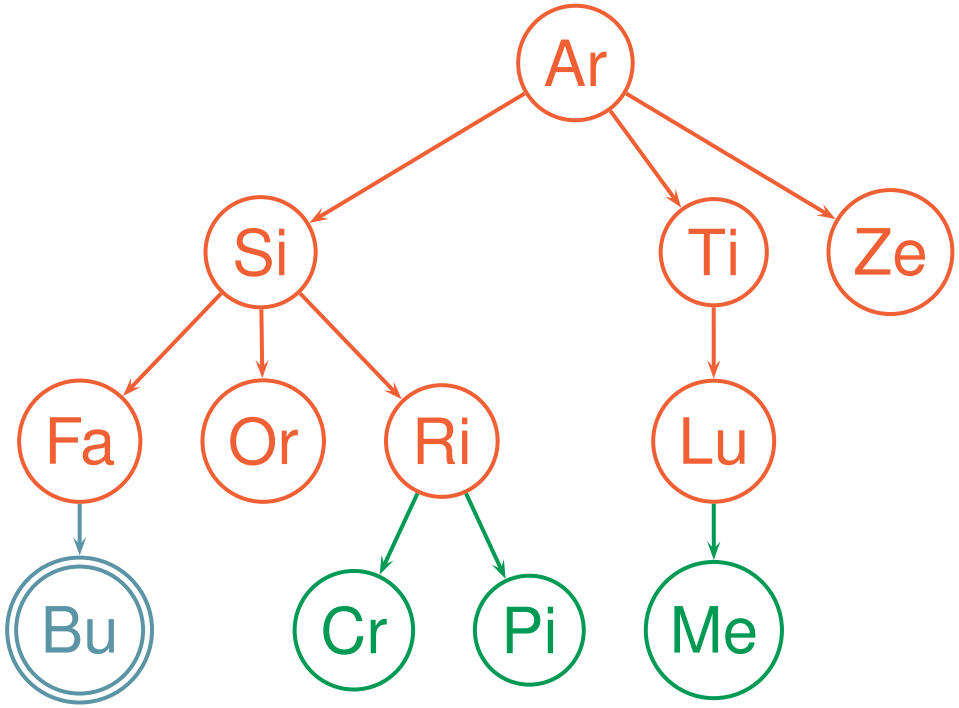
$Append(O, n)$ // n added to the queue

return NULL // no solution found

- ***Question 1***: Write a trace of the ***BFS*** algorithm applied to the problem of finding the shortest path from Arad to Bucarest.

O	C	s
{Arad}	{}	–
{Sibiu, Timisoara, Zerind}	{Arad}	Arad
{Timisoara, Zerind, Fagaras, Oradea, Rimnicu}	{Arad, Sibiu}	Sibiu
{Zerind, Fagaras, Oradea, Rimnicu, Lugoj}	{Arad, Sibiu, Timisoara}	Timisoara
{Fagaras, Oradea, Rimnicu, Lugoj}	{Arad, Sibiu, Timisoara, Zerind}	Zerind
{Oradea, Rimnicu, Lugoj, Bucharest}	{Arad, Sibiu, Timisoara, Zerind, Fagaras}	Fagaras
{Rimnicu, Lugoj, Bucharest}	{Arad, Sibiu, Timisoara, Zerind, Fagaras, Oradea}	Oradea
{Lugoj, Bucharest, Craiova, Pitesti}	{Arad, Sibiu, Timisoara, Zerind, Fagaras, Oradea, Rimnicu}	Rimnicu
{Bucharest, Craiova, Pitesti, Mehadia}	{Arad, Sibiu, Timisoara, Zerind, Fagaras, Oradea, Rimnicu, Lugoj}	Lugoj
{Craiova, Pitesti, Mehadia}	{Arad, Sibiu, Timisoara, Zerind, Fagaras, Oradea, Rimnicu, Lugoj, Bucharest}	Bucharest

- ***Question 2***: Draw the search tree as a result of applying the ***BFS*** algorithm to the problem of finding the shortest path from Arad to Bucarest.



References

[1] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, third edition, 2010.

[2] E. Moore. The shortest path through a maze. In *Proc. of the Int. Symposium on the Theory of Switching, Part II*, pages 285–292. Harvard University Press, 1959.

[3] C. Y. Lee. An algorithm for path connections and its applications. *IRE Trans. on Electronic Computers*, EC-10, 1961.

[4] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, 2018.