UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Workbook:

Uniform-cost search: Dijkstra’s algorithm

Albert Sanchis
Jorge Civera

DSIC

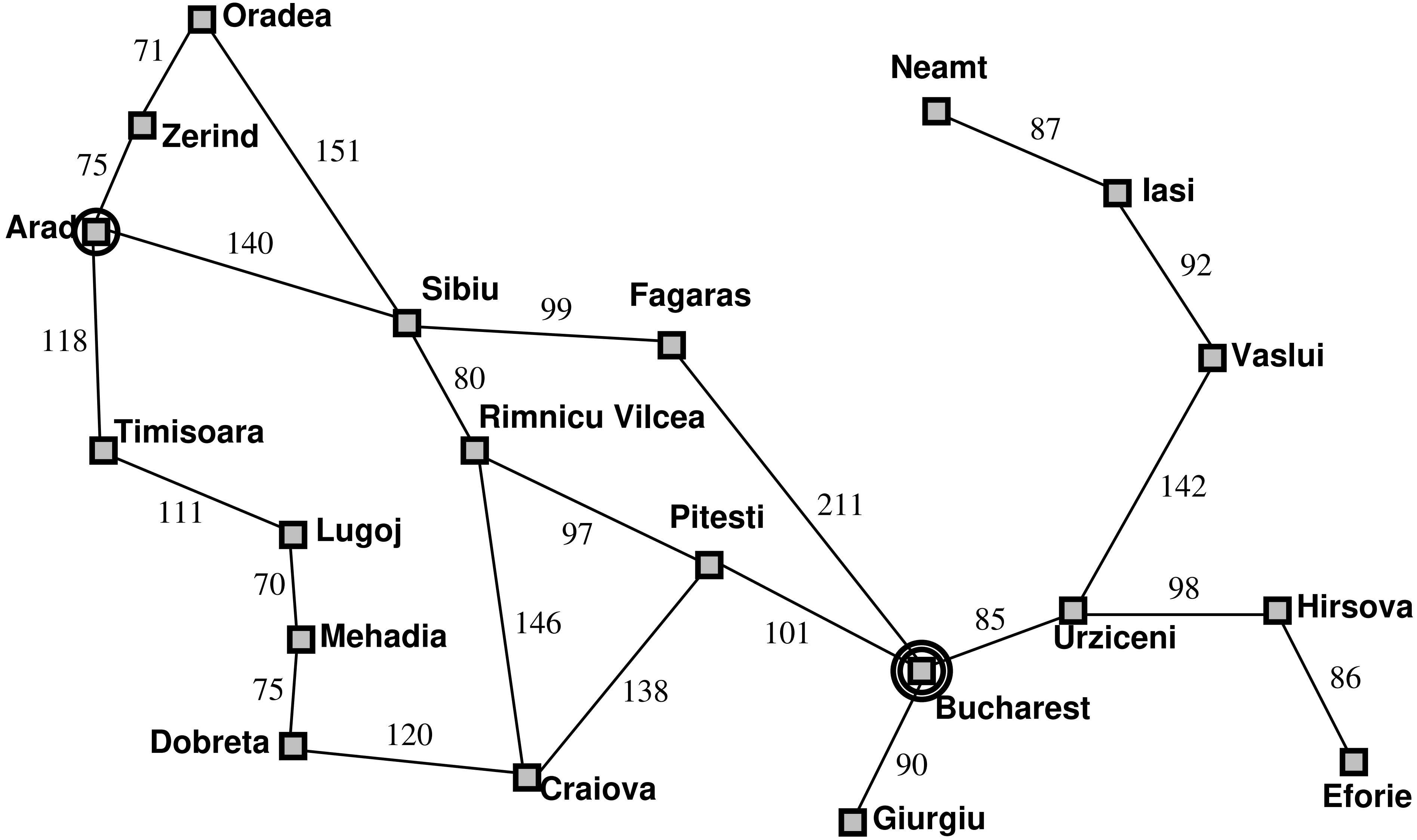
Departament de Sistemes
Informàtics i Computació

Learning objectives

- To describe uniform-cost search or Dijkstra's algorithm.
- To draw a uniform-cost search tree.
- To apply uniform-cost search to a well-known problem
- To analyze the quality of uniform-cost search.

Problem: Shortest path between two points

Shortest path from Arad to Bucarest [1]:



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

Uniform-cost or Dijkstra's algorithm [1, 2, 3]

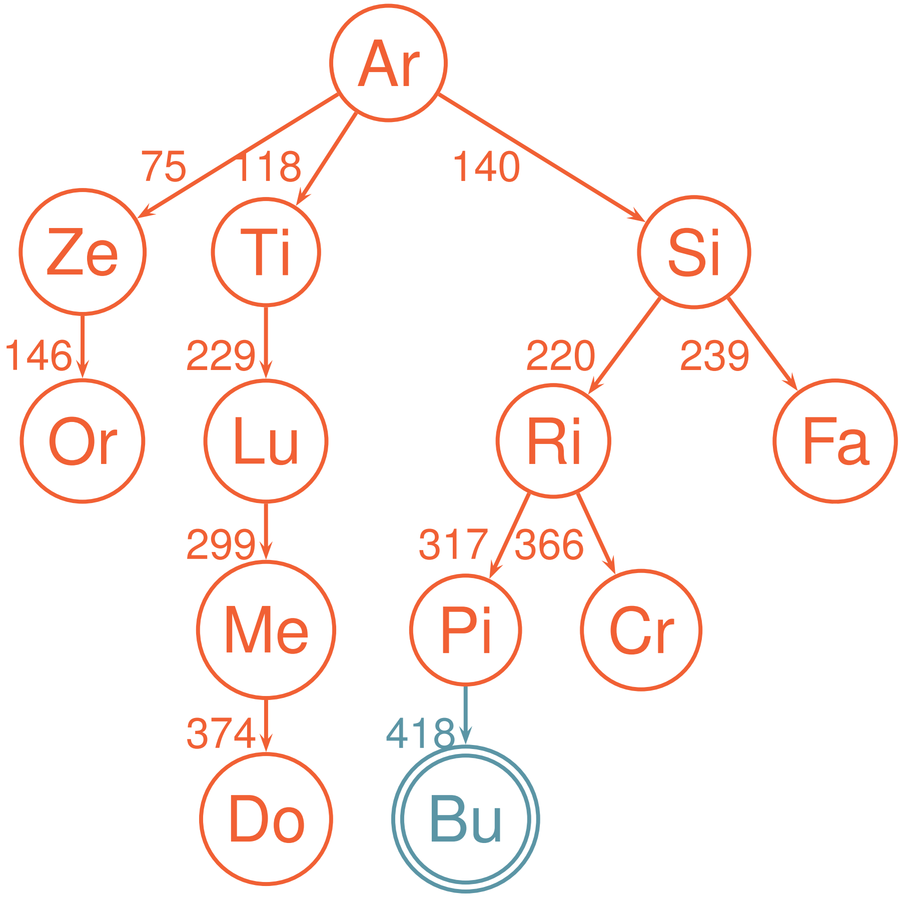
```
UCS($G, s'$)                         //Uniform-cost search; $G$ weighted graph, $s'$ start
  $O = InitQueue(s', g_{s'} \triangleq 0)$          // Open: priority queue $g$
  $C = \emptyset$                                   // Closed: explored nodes
  while not $EmptyQueue(O)$:          // best-first: $s = \arg\min_{n \in O} g_n$
    $s = Pop(O)$                                // ties solved in favor of goals
    if $Goal(s)$ return $s$                              // solution found!
    $C = C \cup \{s\}$                                       // $s$ explored
    forall $(s, n) \in Adjacents(G, s)$:            // generation: $n$ child of $s$
      $x = g_s + w(s, n)$                // path cost from $s'$ to $n$ through $s$
      if                  $n \notin C \cup O$: $Push(O, n, g_n \triangleq x)$
      else if $n \in O$ and $x < g_n$: $Update(O, n, g_n \triangleq x)$
  return NULL                                       // no solution found
```

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

O	C	s
{Arad (c=0)}	{}	–
{Zerind (c=75), Timisoara (c=118), Sibiu (c=140)}	{Arad (c=0)}	Arad (c=0)
{Timisoara (c=118), Sibiu (c=140), Oradea (c=146)}	{Arad (c=0), Zerind (c=75)}	Zerind (c=75)
{Sibiu (c=140), Oradea (c=146), Lugoj (c=229)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118)}	Timisoara (c=118)
{Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140)}	Sibiu (c=140)
{Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146)}	Oradea (c=146)
{Lugoj (c=229), Fagaras (c=239), Pitesti (c=317), Craiova (c=366)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220)}	Rimnicu (c=220)
{Fagaras (c=239), Mehadia (c=299), Pitesti (c=317), Craiova (c=366)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229)}	Lugoj (c=229)
{Mehadia (c=299), Pitesti (c=317), Craiova (c=366), Bucharest (c=450)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239)}	Fagaras (c=239)

O	C	s
{Pitesti (c=317), Craiova (c=366), Dobreta (c=374), Bucharest (c=450)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239), Mehadia (c=299)}	Mehadia (c=299)
{Craiova (c=366), Dobreta (c=374), Bucharest (c=418)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239), Mehadia (c=299), Pitesti (c=317)}	Pitesti (c=317)
{Dobreta (c=374), Bucharest (c=418)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239), Mehadia (c=299), Pitesti (c=317), Craiova (c=366)}	Craiova (c=366)
{Bucharest (c=418)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239), Mehadia (c=299), Pitesti (c=317), Craiova (c=366), Dobreta (c=374)}	Dobreta (c=374)
{}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239), Mehadia (c=299), Pitesti (c=317), Craiova (c=366), Dobreta (c=374)}	Bucharest (c=418)

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



- ***Question 3***: Does the IDS algorithm find a solution? ***Yes***
- ***Question 4***: If the answer is “Yes”:
  - What is the solution found? ***The solution path is: Arad, Sibiu, Rimnicu, Pitesti, Bucharest***
  - What is the cost of this solution? ***418***
  - Is this the solution of minimum cost? ***Yes***
  - What type of solution is found by the UCS algorithm? ***The optimal solution if the cost of actions are positive***

References

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

[2] E. W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1959.

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