UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Workbook:
Iterative deepening search

Albert Sanchis
Jorge Civera

DSIC

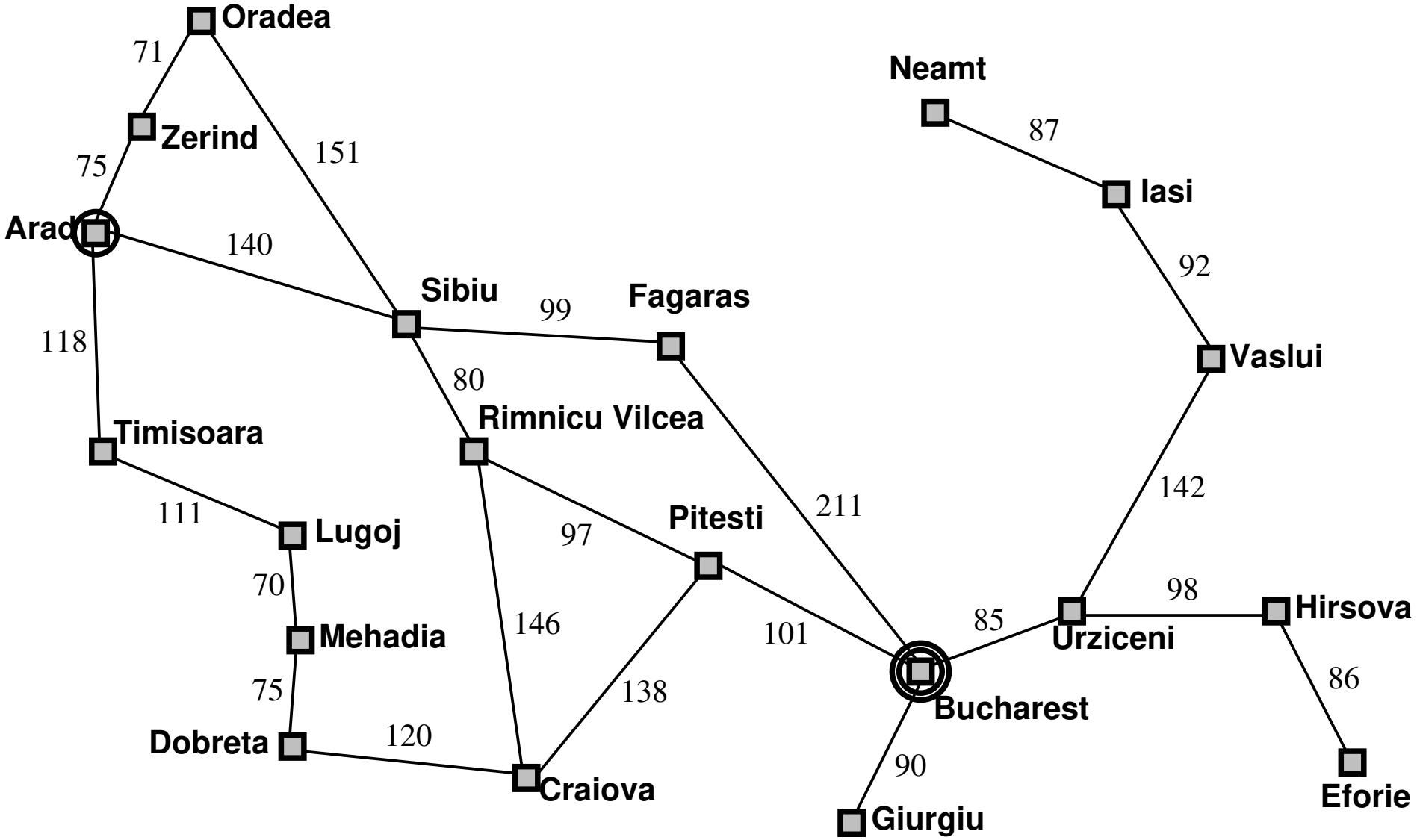
Departament de Sistemes Informàtics i Computació

Learning objectives

- To describe iterative deepening search.
- To draw the trees of iterative deepening search.
- To apply iterative deepening search to a well-known problem
- To analyze the quality of iterative deepening search.

Problem: Shortest path between two points

Shortest path from Arad to Bucarest [1]:



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

Iterative deepening search [2]

IDS(G, s) // ***Iterative deepening search***
  for $m = 0, 1, 2, \ldots$: **if** $(r = \text{DFS}(G, s, m)) \neq \text{NULL}$: **return** r

DFS(G, s', m) // ***Depth-first search*** with maximum depth of m
  $O = InitStack(s')$ // ***Open: search frontier-stack***
  while not $EmptyStack(O)$:
    $s = Pop(O)$ // selection ***LIFO (Last in, first out)***
    if $Goal(s)$ **return** s // solution found!
    if $Depth(s) < m$: // maximum depth not reached
      forall $(s, n) \in Adjacents(G, s)$: // generation: n child of s
        $Push(O, n)$ // n added to the stack
  return NULL // no solution found

- ***Question 1***: Write a trace of the ***IDS*** algorithm applied to the problem of finding the shortest path from Arad to Bucarest.
- ***Question 2***: Draw the search tree as a result of applying the ***IDS*** algorithm to the problem of finding the shortest path from Arad to Bucarest with maximum depth $m = 3$.
- ***Question 3***: Does the IDS algorithm find a solution?
- ***Question 4***: If the answer is "Yes":
  - How many iterations does the IDS algorithm need to find the solution?
  - What the number of iterations depends on?
  - What is the solution found?
  - What is the cost of this solution?
  - Is this the solution of minimum cost?
  - What type of solution is found by the IDS algorithm?

References

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

[2] R. E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 1985.