UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Workbook:
Depth-first search (backtracking)

Albert Sanchis
Jorge Civera

DSIC

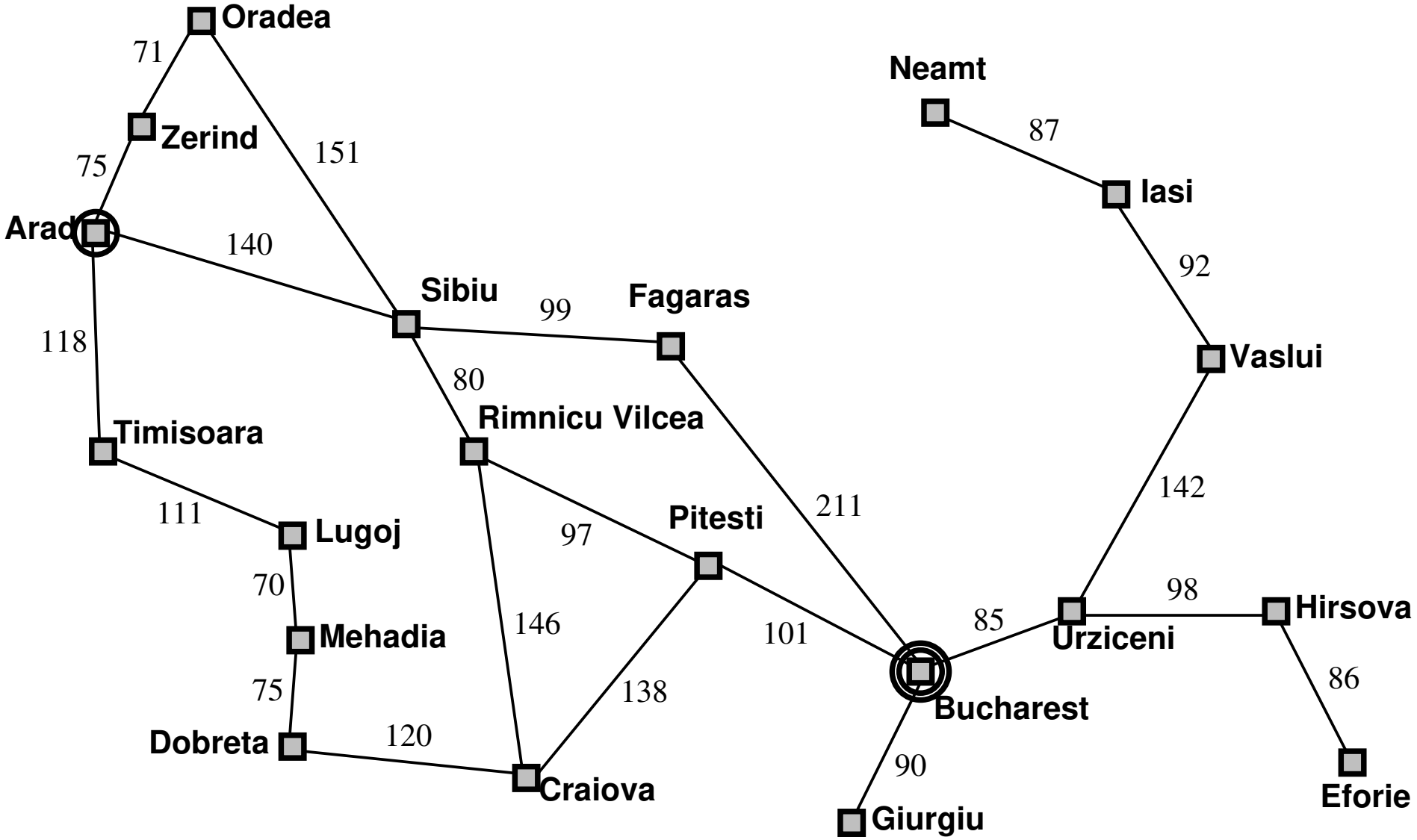
Departament de Sistemes Informàtics i Computació

Learning objectives

- To describe depth-first search (backtracking).
- To draw the tree of depth-first search.
- To apply depth-first search (backtracking) to a well-known problem
- To analyze the quality of depth-first search (backtracking).

Problem: Shortest path between two points

Shortest path from Arad to Bucarest [1]:



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

Backtracking

DFS (recursive) variant with individual child generation:

```
BT(G, s, m)                    // Backtracking with maximum depth of m
  if Goal(s) return s                          // solution found!
  if m = 0 return NULL                         // maximum depth reached
  n = FirstAdjacent(G, s)                      // generation: n first child of s
  while n ≠ NULL:
    r = BT(G, n, m − 1)                        // current child result
    if r ≠ NULL: return r                      // if r is solution, stop
    n = NextAdjacent(G, s, n)                  // generation: n next child of s
  return NULL                                  // no solution found
```

- ***Question 1***: Draw the search tree as a result of applying the ***DFS*** algorithm (backtracking) to the problem of finding the shortest path from Arad to Bucarest with maximum depth $m = 3$.
- ***Question 2***: Does the DFS algorithm (backtracking) find a solution?
- ***Question 3***: If the answer is "Yes":
  - 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 DFS algorithm (backtracking)?
- ***Question 4***: What happens if a maximum depth is not defined?

References

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