



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

# Quadern de treball: Iterative Deepening $A^*$ (IDA $^*$ )<sup>1</sup>

Albert Sanchis

*DSIC*

Departament de Sistemes  
Informàtics i Computació

---

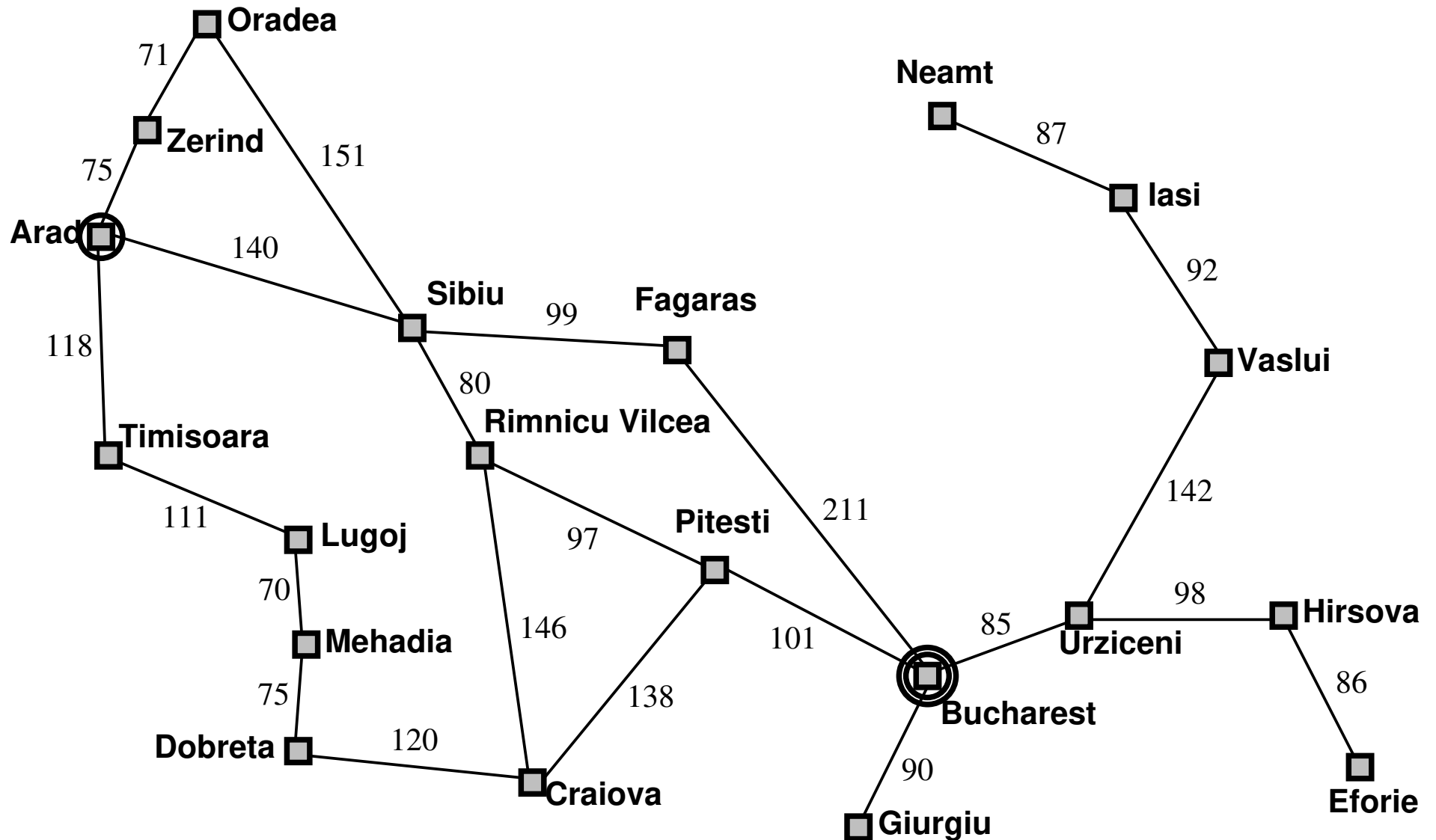
<sup>1</sup>Per a una correcta visualització, es requereix l'Acrobat Reader v. 7.0 o superior

# Objectius formatius

- ▶ Caracteritzar la cerca convencional en un graf d'estats.
- ▶ Descriure cerca IDA\*.
- ▶ Construir l'arbre de cerca IDA\*.
- ▶ Aplicar cerca IDA\* a un problema clàssic.
- ▶ Analitzar la qualitat de cerca IDA\*.

# Problema: La ruta més curta entre dos punts

Cerca d'una ruta més curta des d'Arad a Bucarest [1]:



$\text{Accions}(\text{Arad}) = \{\text{Anar}(\text{Sibiu}), \text{Anar}(\text{Timisoara}), \text{Anar}(\text{Zerind})\}.$

# Problema: La ruta més curta entre dos punts

Distàncies en línia recta a Bucharest

	Bucharest		Bucharest
Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

# L'algorisme IDA\* (main) [2]

```
IDA( $G, s', h$ )                                //  $G$  graf ponderat,  $s'$  inici,  $h$  heurística  
   $P = \text{InitStack}(s')$                         // Inicialitza Path amb el node arrel  
   $b = h(s')$                                     // Inicialitza el límit amb  $f_{s'} = h(s')$   
  while True:  
     $(nextb, r) = \mathbf{BT}(G, P, h, b)$            //  $nextb$  límit següent;  $r$  estat obj.  
    if  $r \neq \text{NULL}$ : return  $P$                 // si solució, torna Path al objectiu  
    if  $nextb = \infty$ : return NULL // no fills per a calcular el seg. límit  
     $b = nextb$                                 // actualització del límit per a la iteració següent
```

# L'algorisme IDA\* (backtracking) [2]

<b>BT</b> ( $G, P, h, b$ )	// $G$ graf ponderat, $P$ <b>Path</b> , $h$ , $b$ límit
$s = Top(P)$	// <b>Path</b> : extrau cim de la pila
$f_s = g_s + h(s)$	// $f$ valor del node a explorar
<b>if</b> $f_s > b$ : <b>return</b> ( $f_s, \text{NULL}$ )	// $b$ excedida fí per a calcular <i>nextb</i>
<b>if</b> $Goal(s)$ : <b>return</b> ( $f_s, s$ )	// solució trobada!
$min = \infty$	// mínim valor d'un fill $f$
$n = FirstAdjacent(G, s)$	// generació: $n$ primer fill de $s$
<b>while</b> $n \neq \text{NULL}$ :	// mentre queden fills per explorar
<b>if</b> $n \notin P$ :	// $n$ no en <b>Path</b> para evitar ciclos
$Push(P, n)$	// afegir fill al <b>Path</b> explorat
( $nextb, r$ ) = <b>BT</b> ( $G, P, h, b$ )	// fill torna mín. $f$ i estat sol.
<b>if</b> $r \neq \text{NULL}$ : <b>return</b> ( $nextb, r$ )	// si $r$ solució, fí recursió
<b>if</b> $nextb < min$ : $min = nextb$	// actualitza valor mín. $f$
$Pop(P)$	// Descarta últim fill de <b>Path</b>
$n = NextAdjacent(G, s, n)$	// generació: $n$ següent fill de $s$
<b>return</b> ( $min, \text{NULL}$ )	// sol. no trobada, torna mínim $f$

- **Qüestió 1:** Construeix l'arbre de cerca resultant d'aplicar l'algorisme **IDA\*** al problema de cerca d'una ruta més curta des d'Arad a Bucarest.

Iteració 1:

## Iteració 2:



## Iteració 3:

## Iteració 4:

## Iteració 5:

## Iteració 6:

- ▶ **Qüestió 2:** L'algorisme troba solució? **Sí**
- ▶ **Qüestió 3:** Si la resposta es “Sí”:
  - ▷ Quina ha sigut la solució trobada? ***El camí solució trobat ha sigut: Arad, Sibiu, Rimnicu, Pitesti, Bucharest***
  - ▷ Quin és el cost d'aquesta solució? **418**
  - ▷ Es tracta de la solució òptima? **Sí**

# Referències

- [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*, 27:97–109, 1985.