

# Inteligencia Artificial

**Oscar Bedoya**

`oscar.bedoya@correounivalle.edu.co`

- \* Algoritmos de búsqueda informada
- \* Heurísticas
- \* Búsqueda Avara
- \* Algoritmo A\*



Un lamento de pájaro,  
Esta vez, muy de mañana,  
En el fondo del barranco.

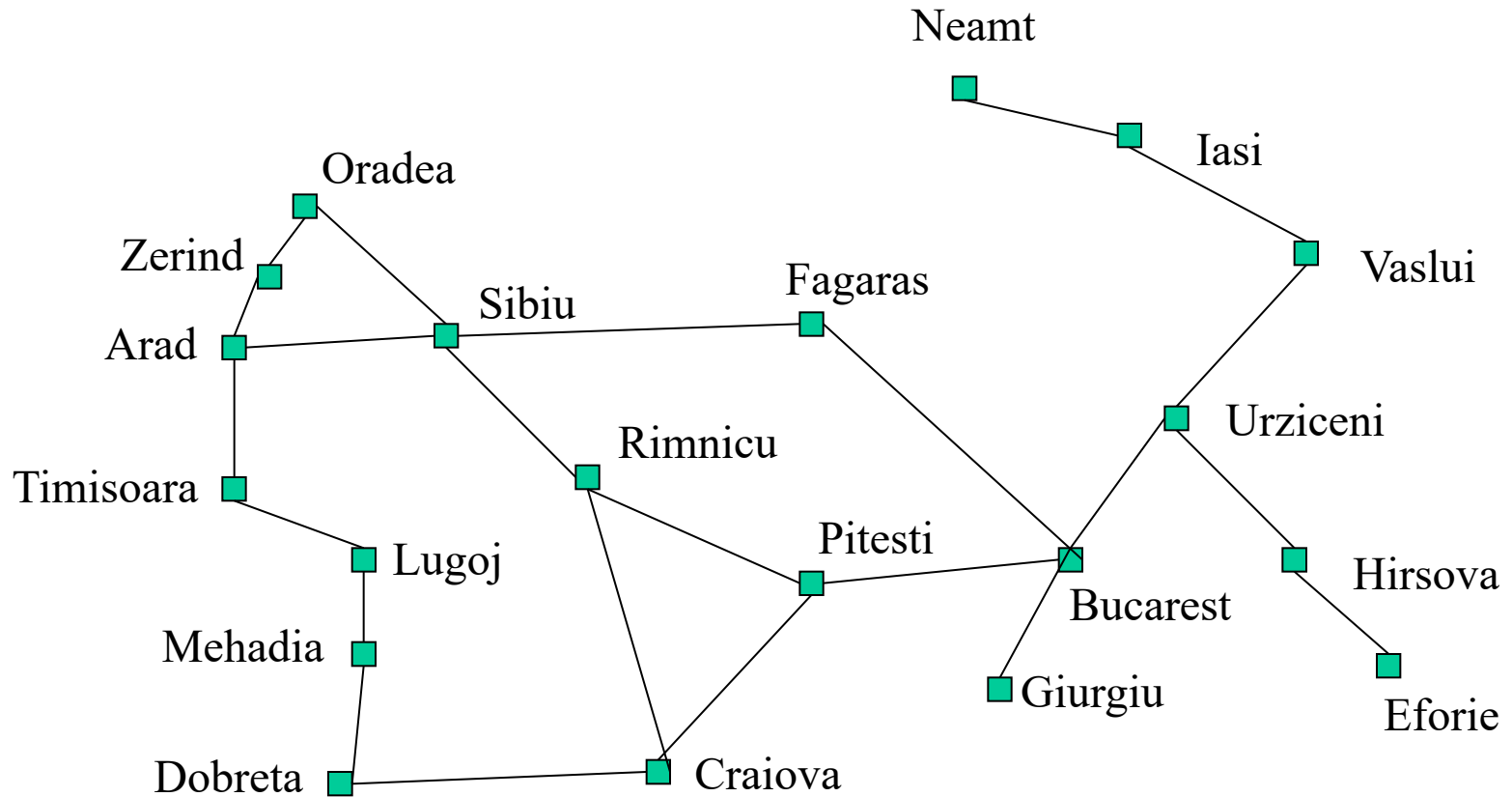
# Búsqueda informada

---

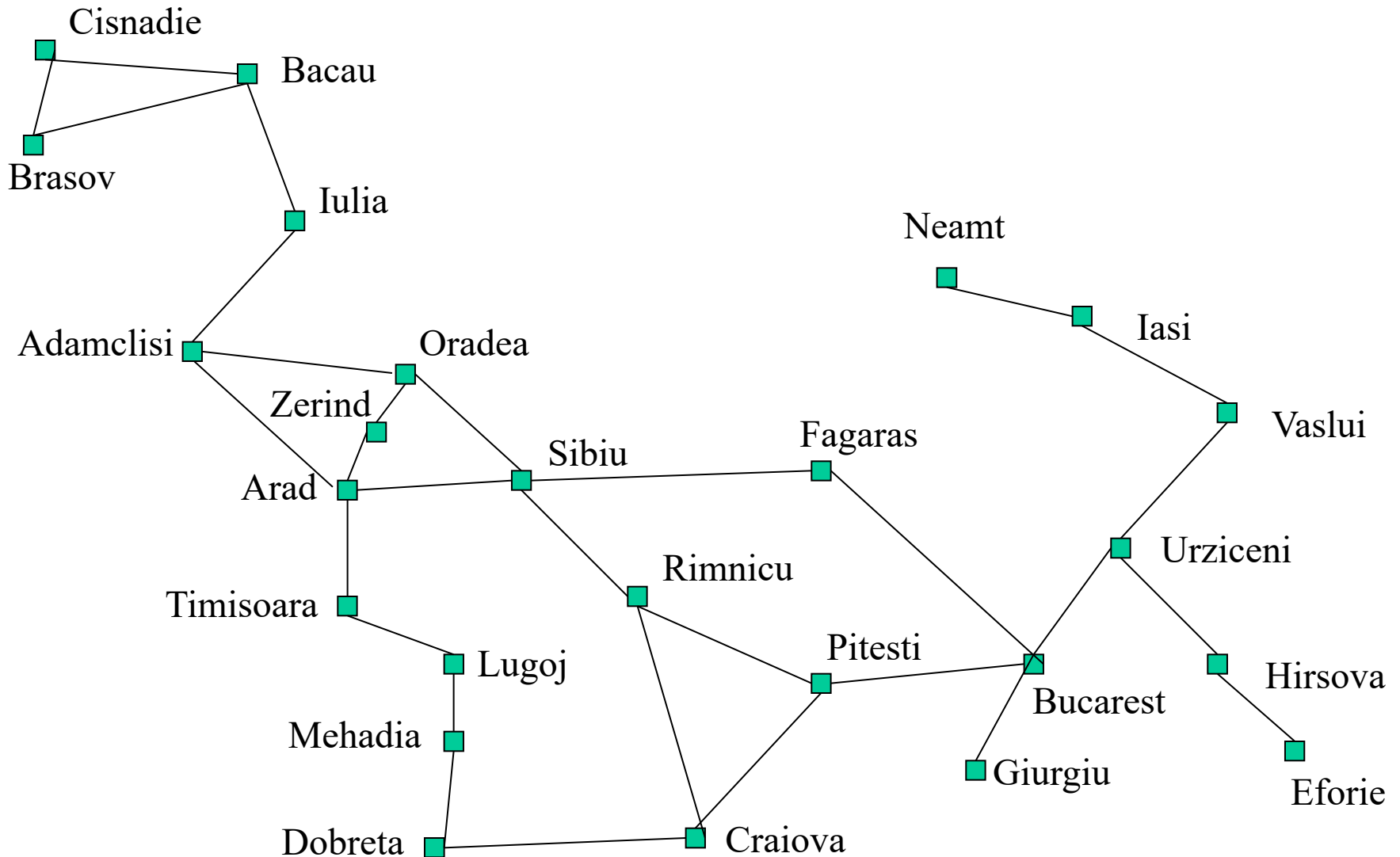
- **Búsqueda no informada:**
  - Costo en tiempo
  - Costo en espacio

# Búsqueda informada

---



# Búsqueda informada



# Búsqueda informada

---

- Búsqueda no informada:
  - Costo en tiempo
  - Costo en espacio
- ¿Cómo mejorar la búsqueda?
  - Seleccionar mejor entre los nodos de un árbol

# Búsqueda informada

---

## Función heurística

- Una heurística es una función que asigna a cada nodo un valor que corresponde al costo estimado de llegar a la meta estando en dicho nodo
- Se denota por  $h(n)$



# Búsqueda informada

---

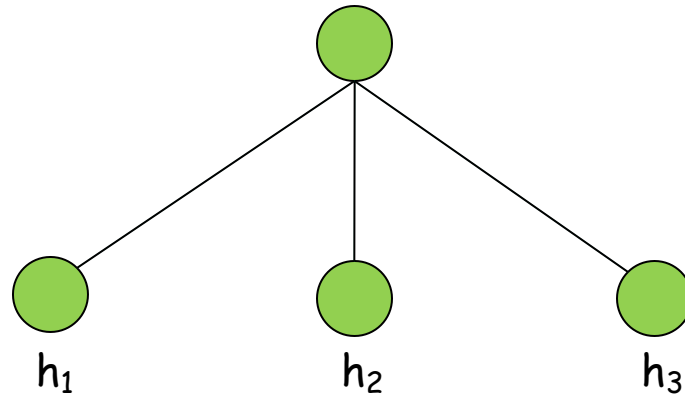
## Función heurística

- Una heurística es una función que asigna a cada nodo un valor que corresponde al costo estimado de llegar a la meta estando en dicho nodo
- Se denota por  $h(n)$

*La heurística estima el valor real,  
no es 100% seguro*

# Búsqueda informada

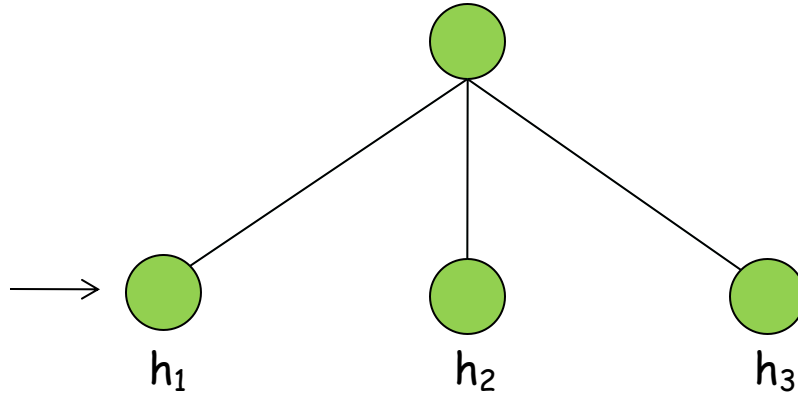
---



# Búsqueda informada

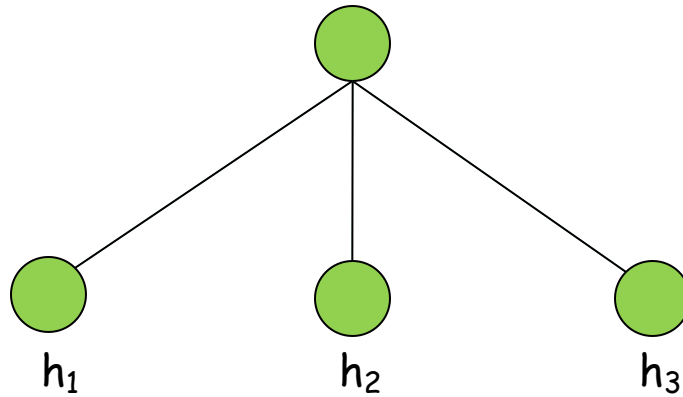
---

$h_1$ : costo estimado  
de llegar a la meta  
desde este nodo



# Búsqueda informada

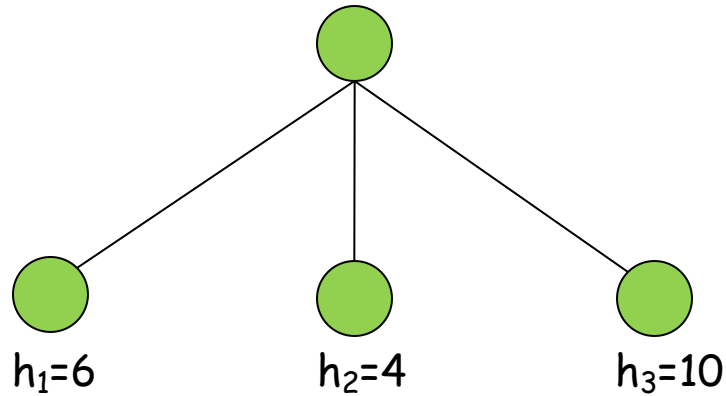
---



Si conociera los valores de  $h$ , cuál de los tres nodos escogería

# Búsqueda informada

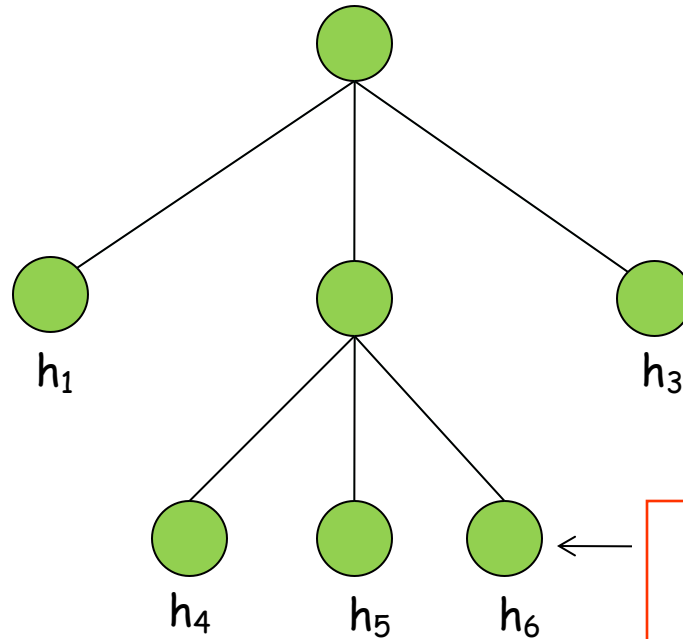
---



Si conociera los valores de  $h$ , cuál de los tres nodos escogería

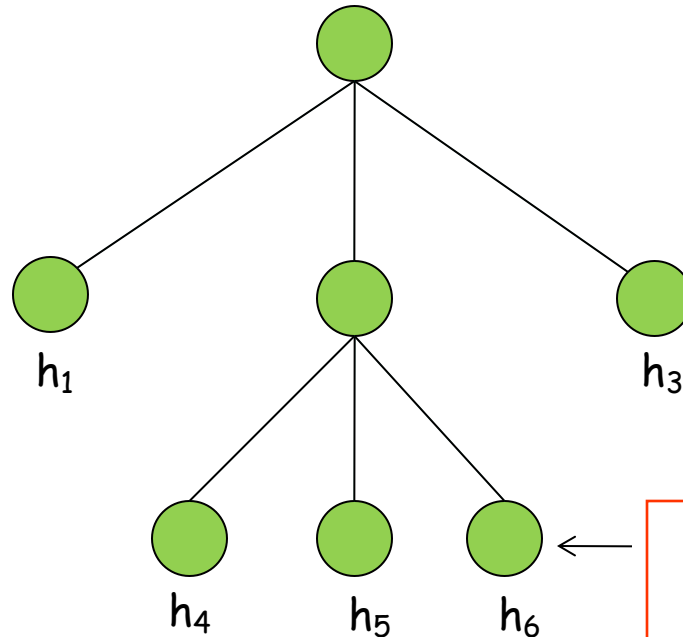
# Búsqueda informada

---



Suponga que se llega a un nodo meta, cuál debería ser el valor de la heurística en este nodo?

# Búsqueda informada

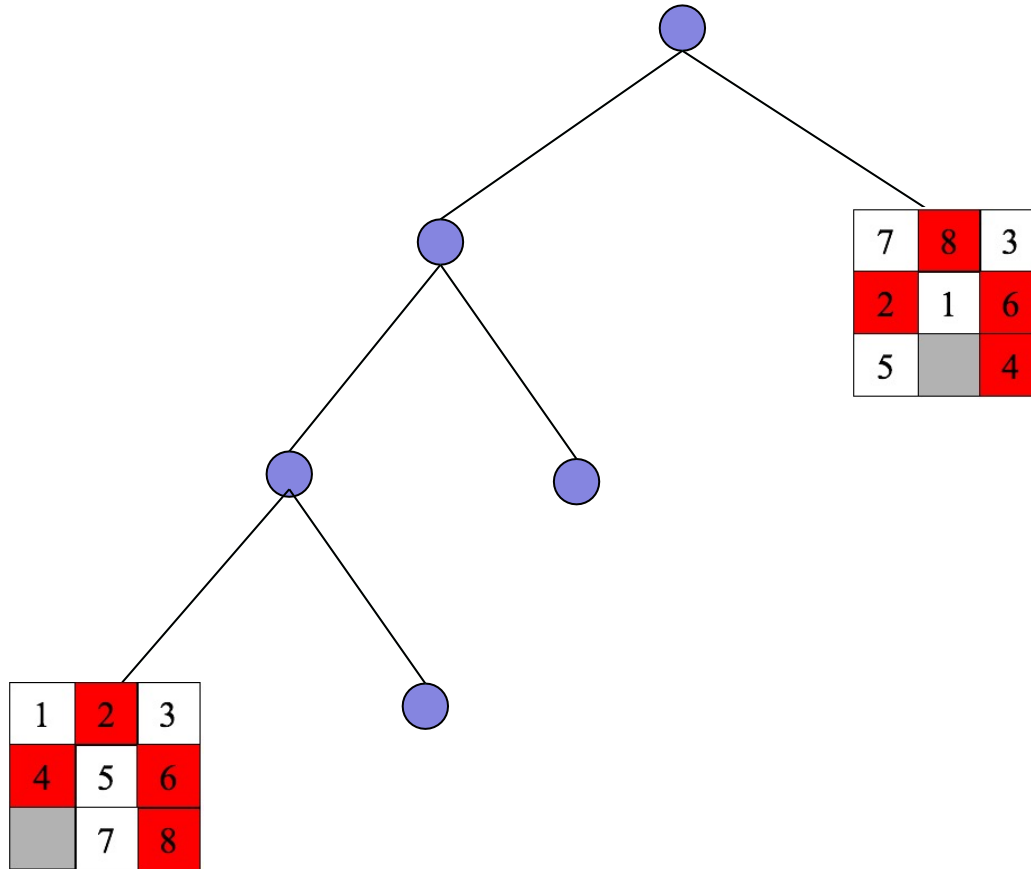


Suponga que se llega a un nodo meta, cuál debería ser el valor de la heurística en este nodo?

La heurística  $h(n)$  debe ser 0 cuando  $n$  es un nodo meta

# Búsqueda informada

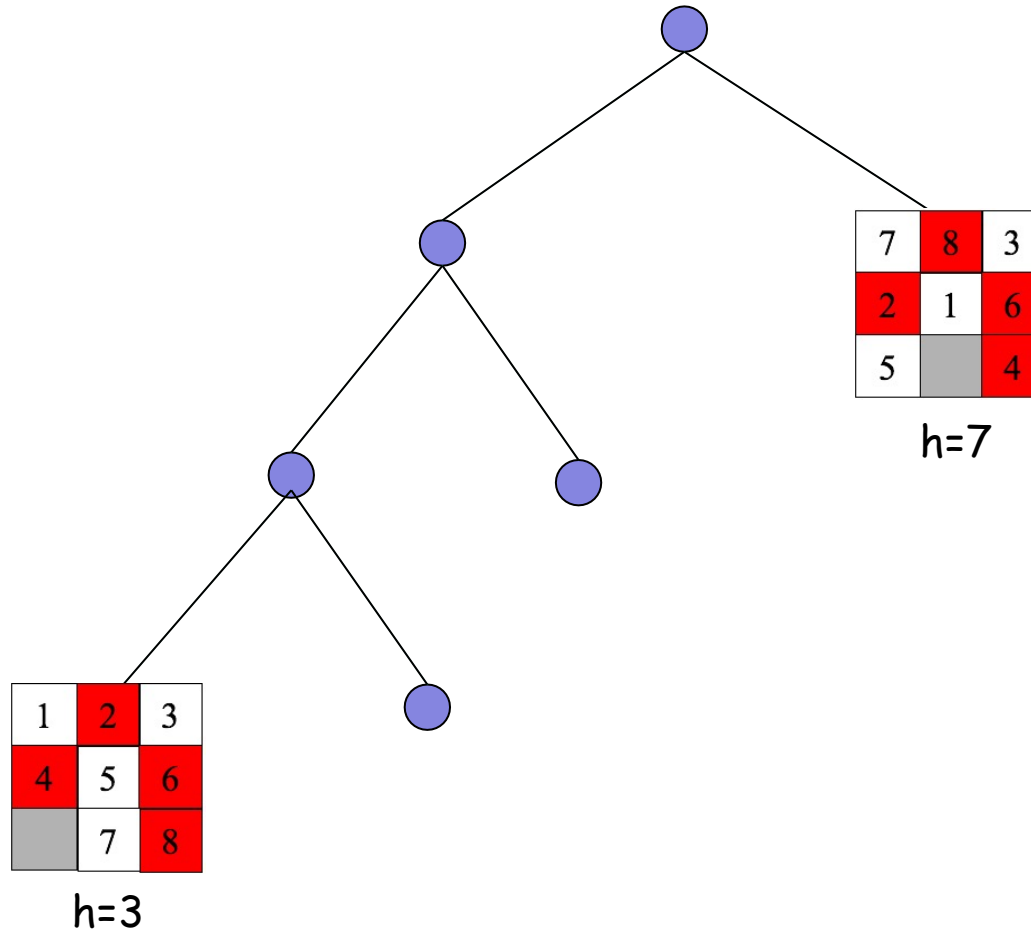
---





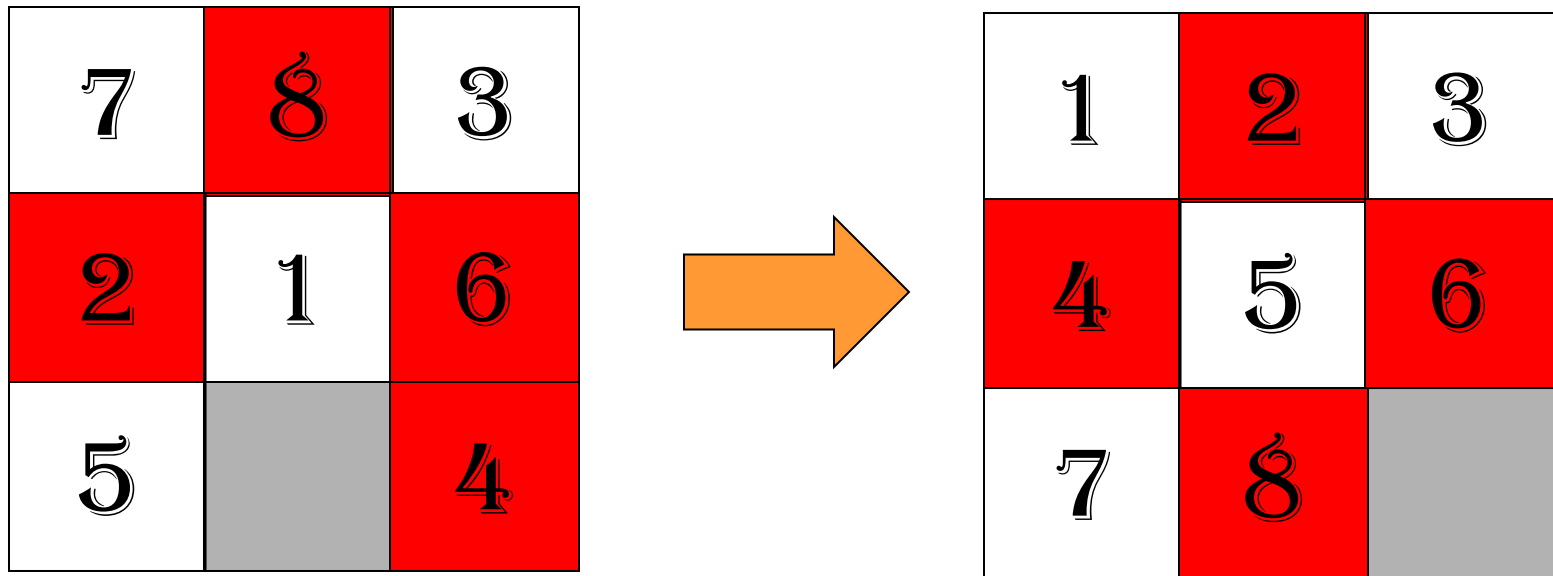
# Búsqueda informada

---



# Búsqueda informada

## 8-puzzle



¿Cuál podría ser una heurística?

# Búsqueda informada

## 8-puzzle

7	8	3
2	1	6
5		4

1	2	3
4	5	6
	7	8

¿Cuál está más distante de la meta?  
¿Cómo lo sabe?

# Búsqueda informada

## 8-puzzle

7	8	3
2	1	6
5		4

1	2	3
4	5	6
	7	8

1	2	3
4	5	6
7	8	

¿Cuál está más distante de la meta? ¿Cómo lo sabe?  
¿Cuál es el valor de la heurística en la meta?

# Búsqueda informada

## 8-puzzle

7	8	3
2	1	6
5		4

1	2	3
4	5	6
	7	8

1	2	3
4	5	6
7	8	

$h(n)$ : cantidad de placas en la posición incorrecta

# Búsqueda informada

## 8-puzzle

7	8	3
2	1	6
5		4

$h(n)=?$

1	2	3
4	5	6
	7	8

$h(n)=?$

1	2	3
4	5	6
7	8	

$h(n)=?$

# Búsqueda informada

## 8-puzzle

7	8	3
2	1	6
5		4

$h(n)=7$

1	2	3
4	5	6
	7	8

$h(n)=3$

1	2	3
4	5	6
7	8	

$h(n)=0$

# Búsqueda informada

---

## 8-puzzle

5	8	7
2	1	4
3		6

1	2	
4	5	6
7	8	3

¿Cuál es de estos nodos escogería para expandir en un árbol de búsqueda?



# Búsqueda informada

## 8-puzzle

5	8	7
2	1	4
3		6

$h(n)=?$

1	2	
4	5	6
7	8	3

$h(n)=?$

# Búsqueda informada

---

## 8-puzzle

5	8	7
2	1	4
3		6

$h(n)=9$

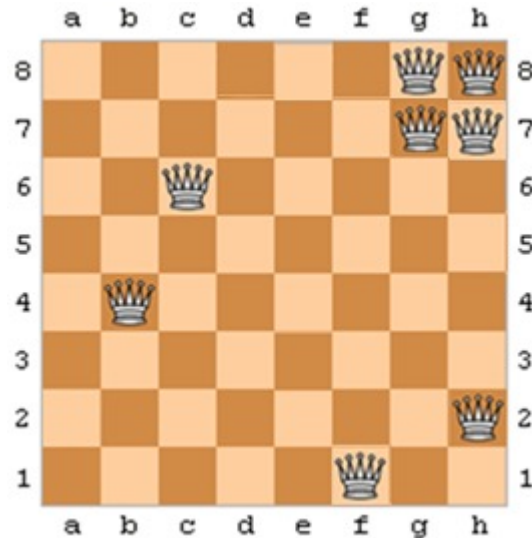
1	2	
4	5	6
7	8	3

$h(n)=2$

# Búsqueda informada

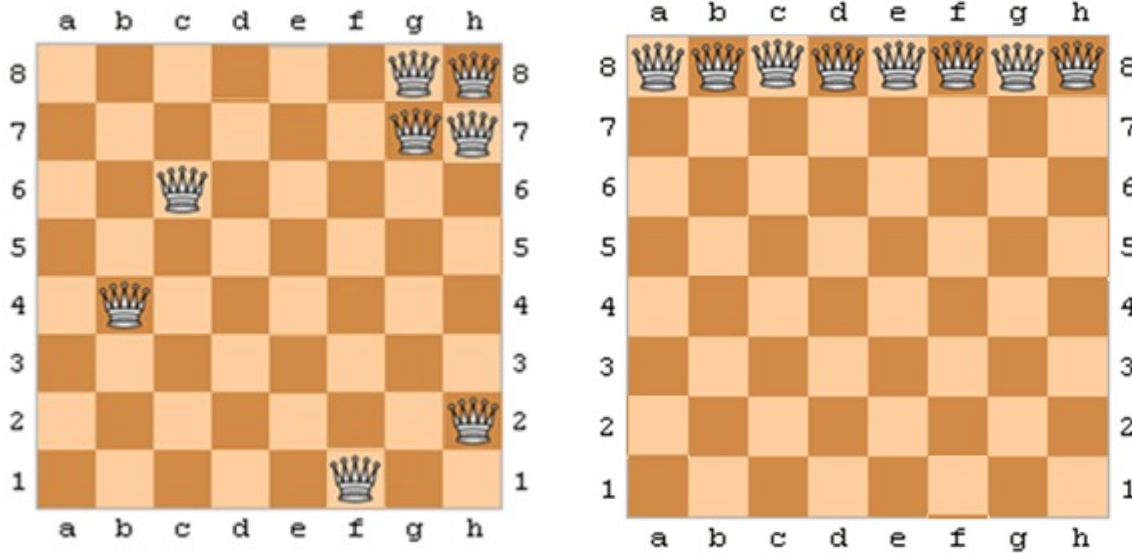
---

- Defina una **heurística**



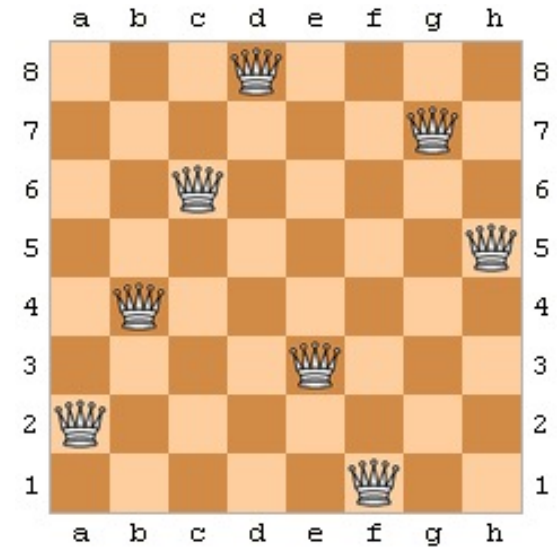
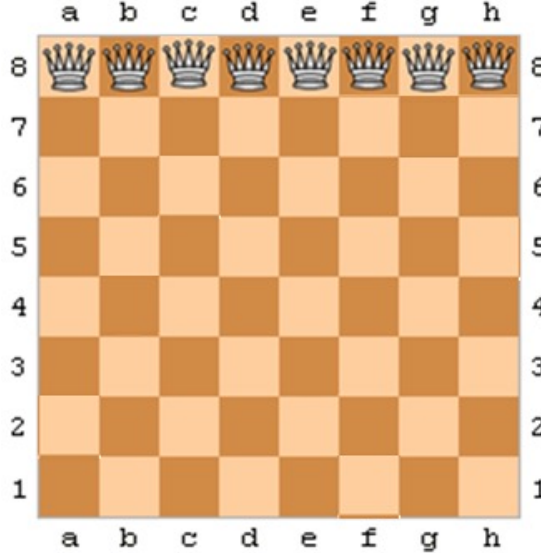
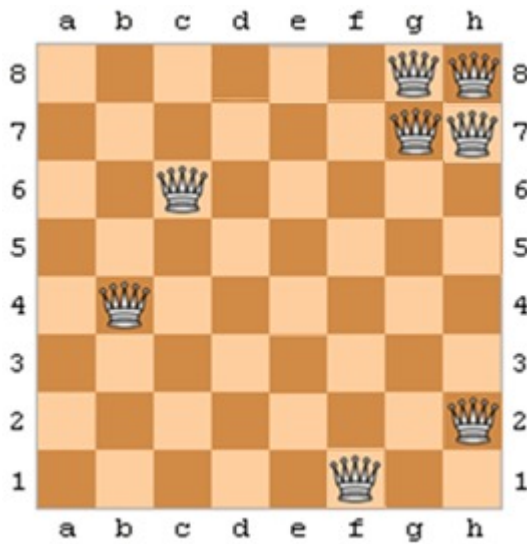
*El problema de las  $n$  reinas*

# Búsqueda informada



¿Cuál está más distante de la meta?  
¿Cómo lo sabe?

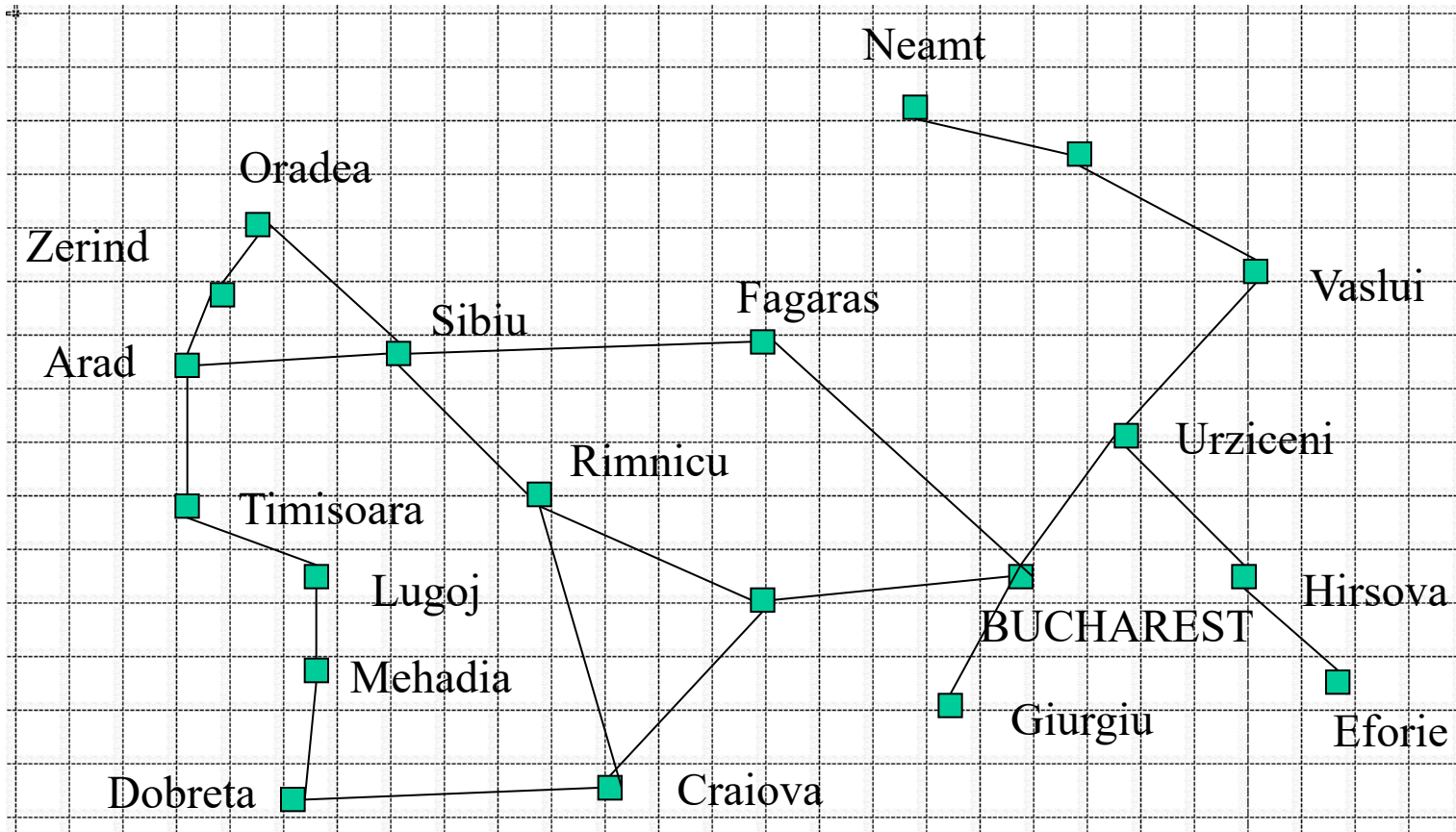
# Búsqueda informada



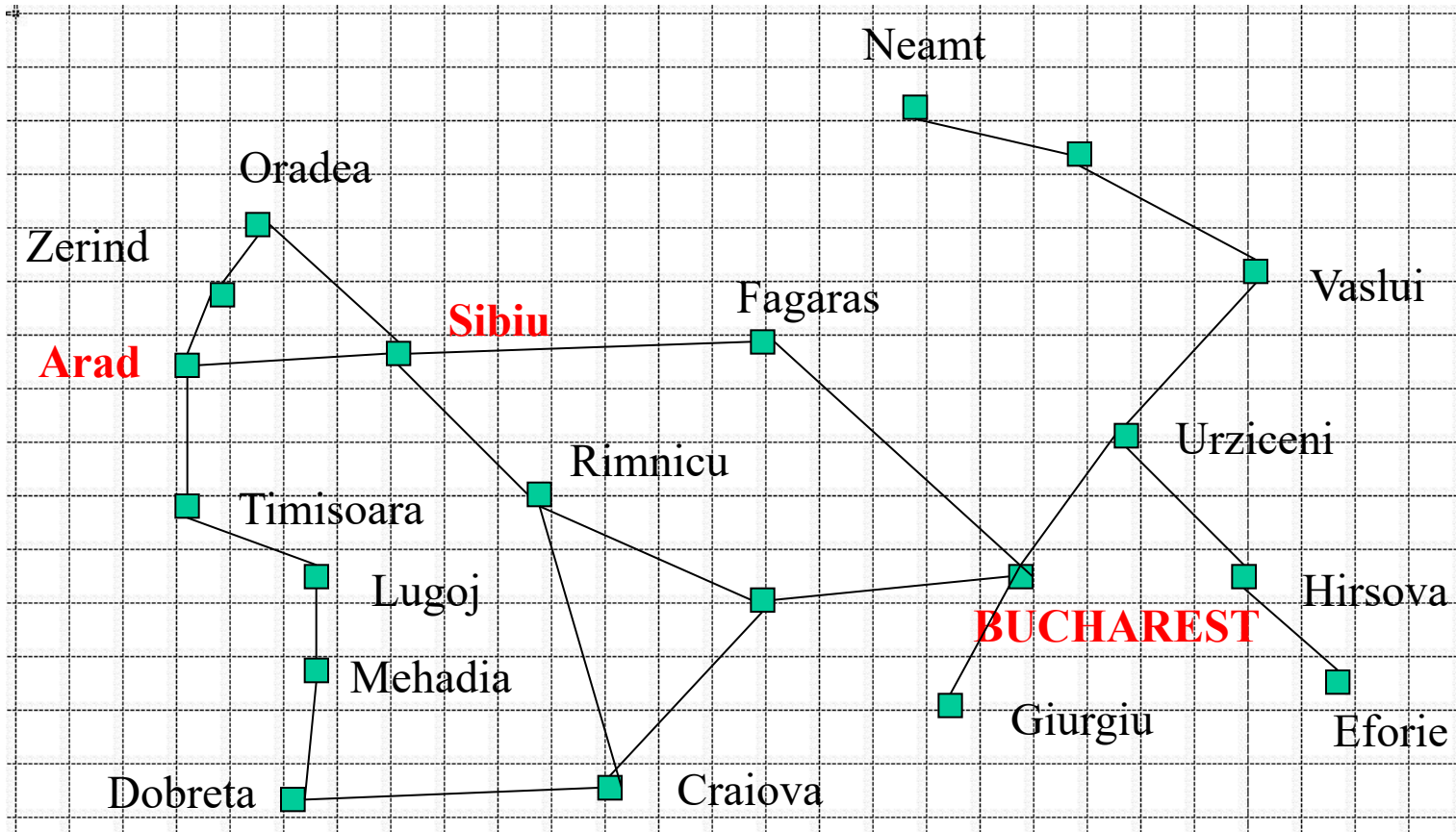
$h(n)$ : cantidad de pares de reinas que se atacan

¿Cuál podría ser una heurística?  
Se cuenta con las coordenadas de las ciudades

# Búsqueda informada



# Búsqueda informada



Ciudad	Coordenadas
Arad	(3,7)
Bucarest	(19,11)
Sibiu	(7,6)

Para llegar a Bucarest, qué es mejor,  
estar en Arad o en Sibiu?



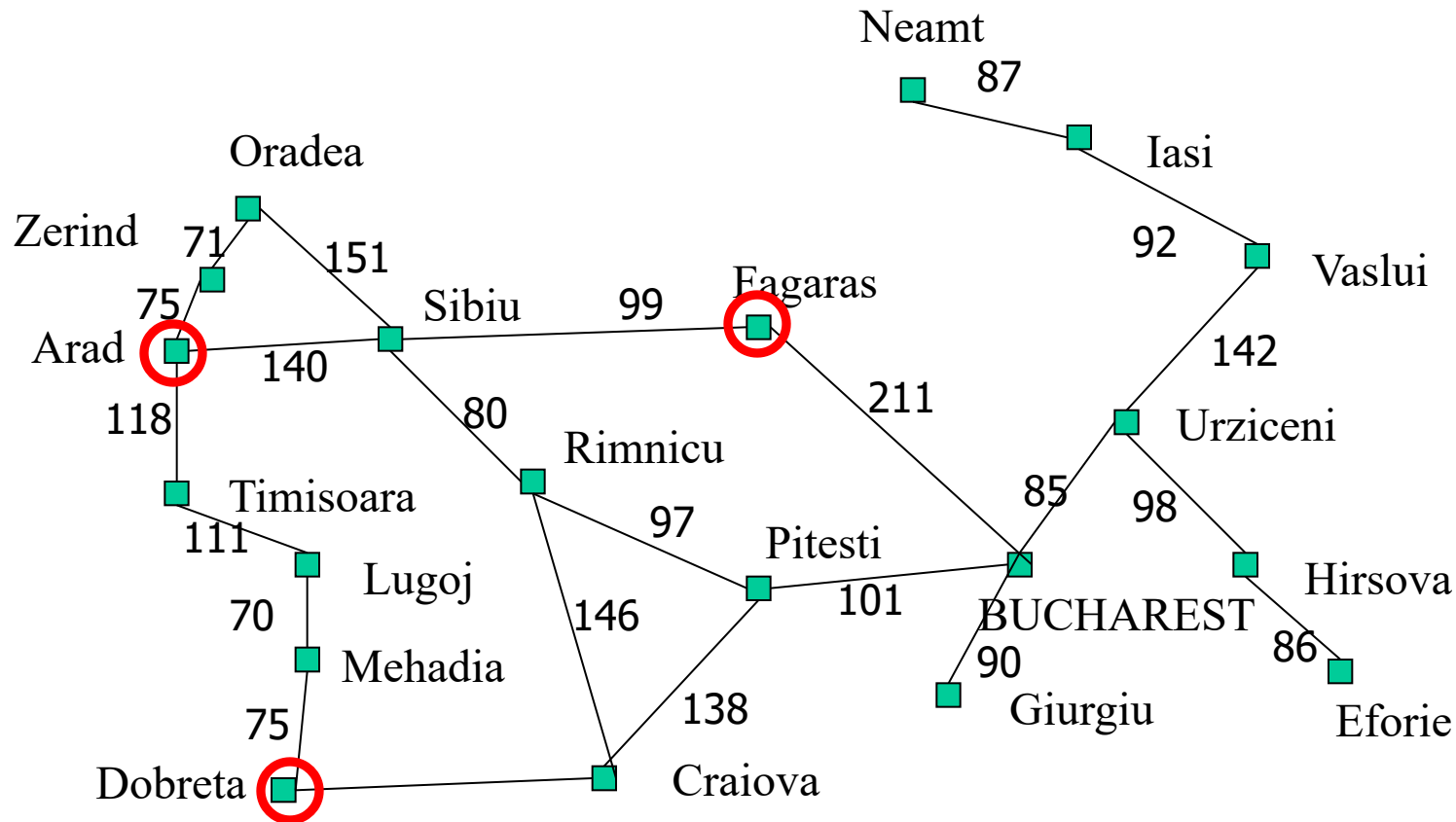
# Búsqueda informada

---

Ciudad	Coordenadas
Arad	(3,7)
Bucarest	(19,11)
Calih	(18,11)
Sibiu	(7,6)

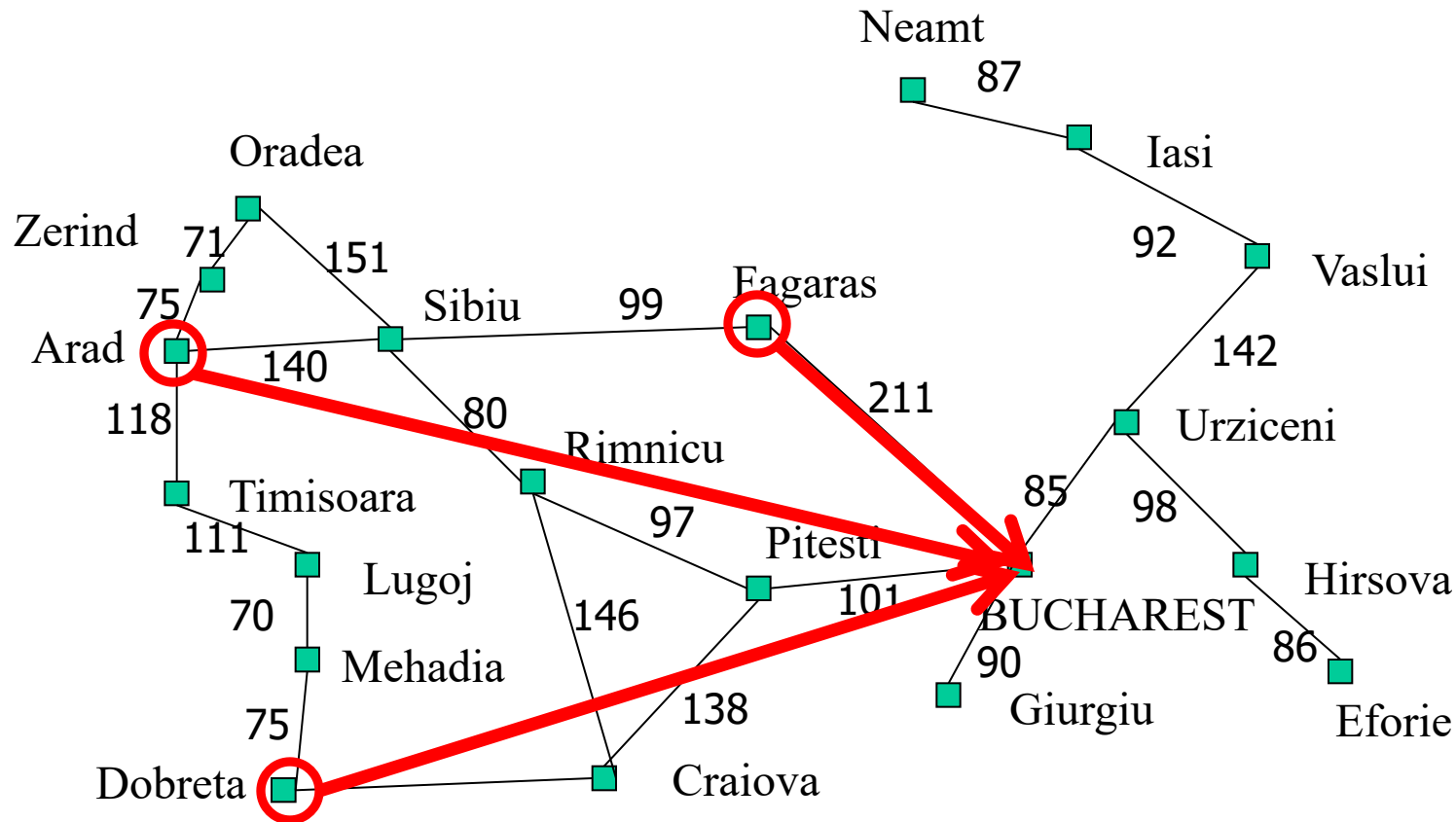
Para llegar a Bucarest, qué es mejor,  
estar en Arad, Calih o Sibiu?

# Búsqueda informada



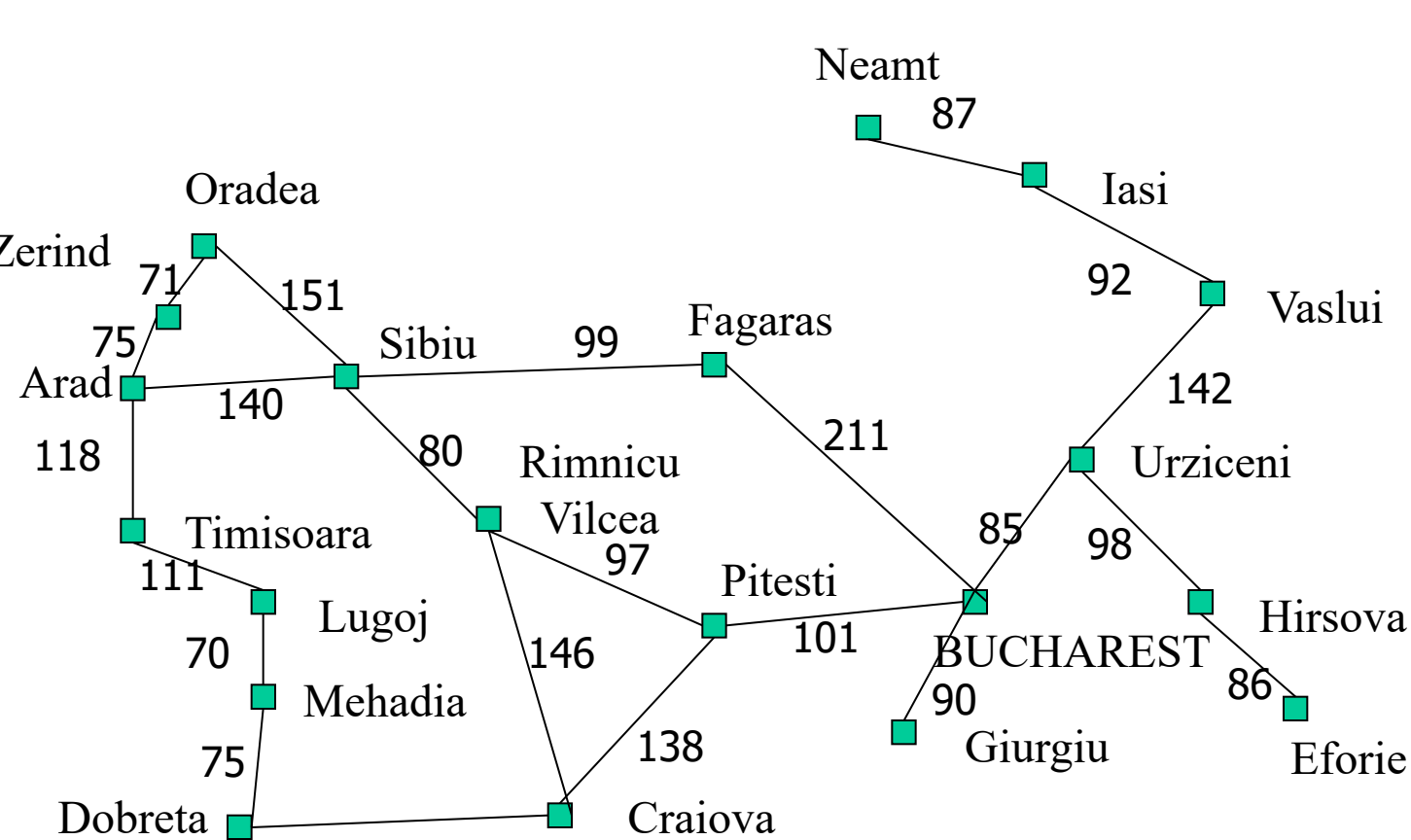
¿Cuál está más distante de la meta? ¿Cómo lo sabe?

# Búsqueda informada



$h(n)$ : distancia en línea recta entre  $n$  y la ubicación de la meta

# Búsqueda informada



Distancia en  
línea recta a  
Bucarest:

Arad	366
Bucarest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

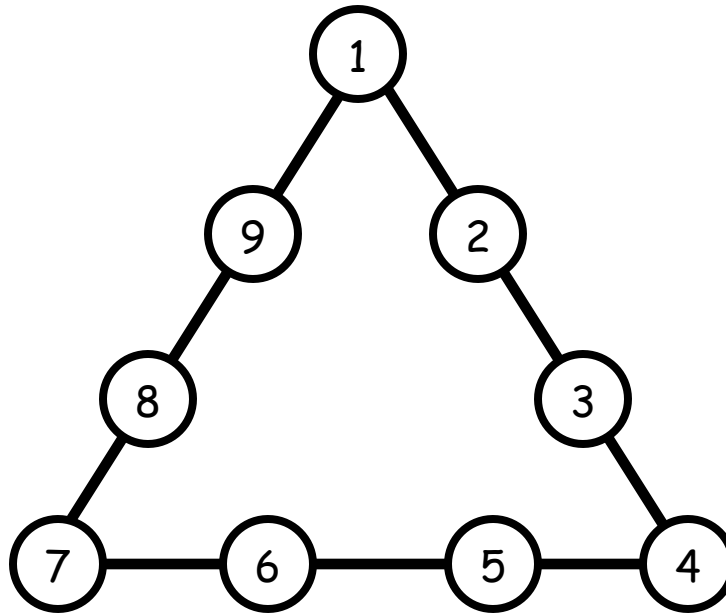
$h(n)$ : distancia en línea recta entre  $n$  y la ubicación de la meta

# Búsqueda informada

---

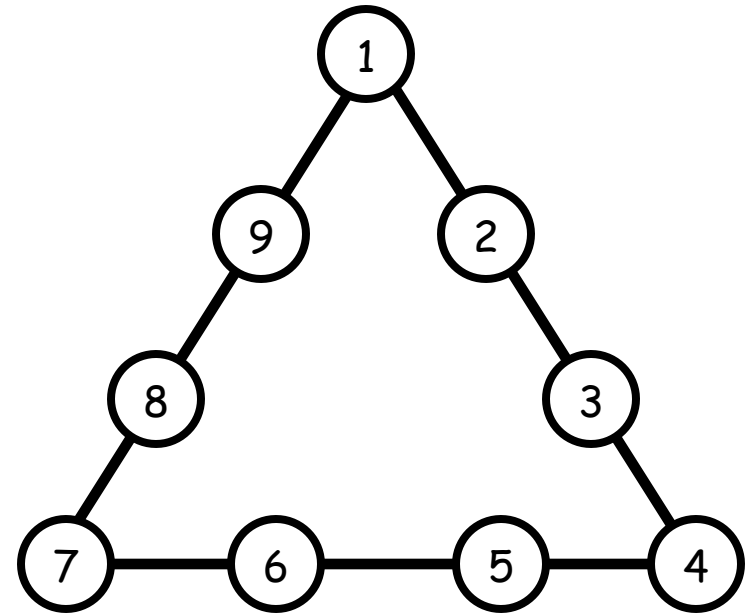
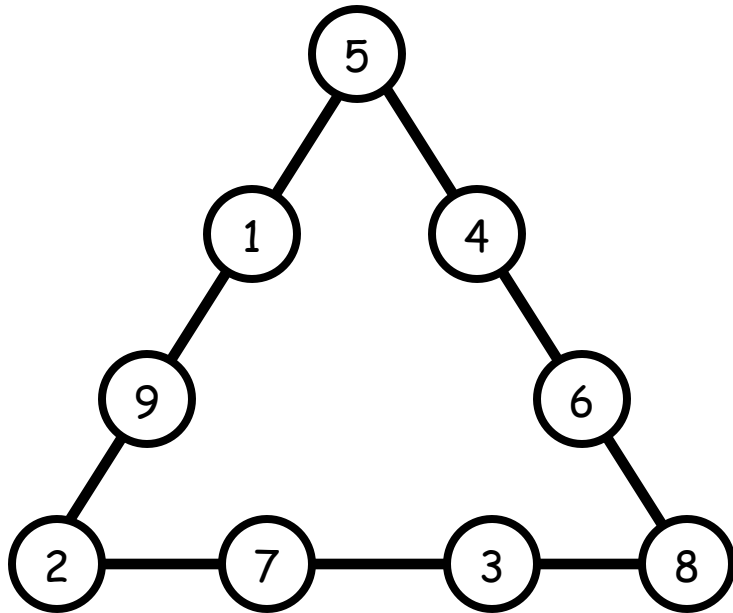
- Defina una **heurística**

**El triángulo mágico.** Utilizando el operador *intercambio(a,b)* que cambia de posición los números a y b, colocarlos de tal forma que la suma sobre cada lado sea 20.



# Búsqueda informada

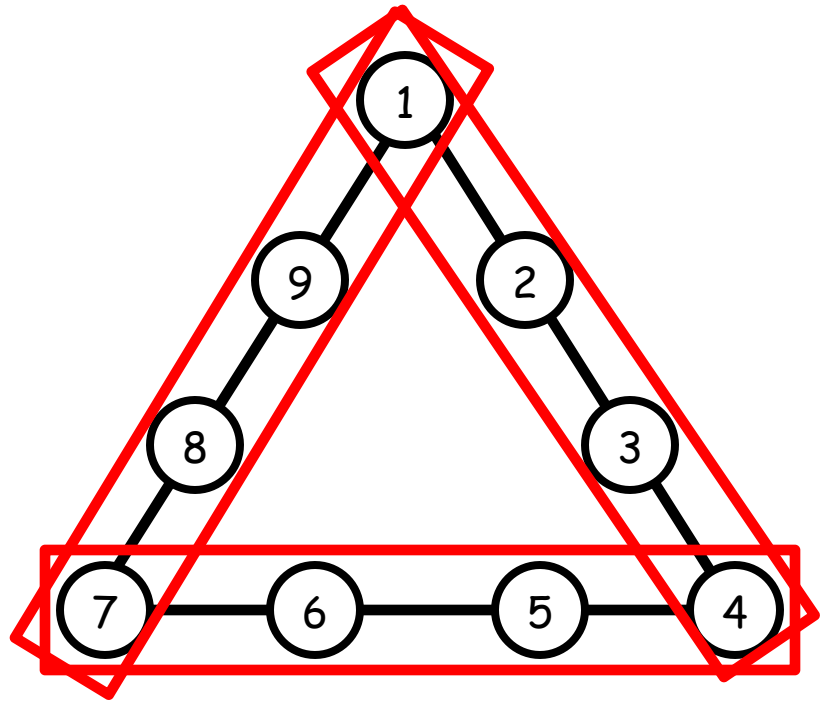
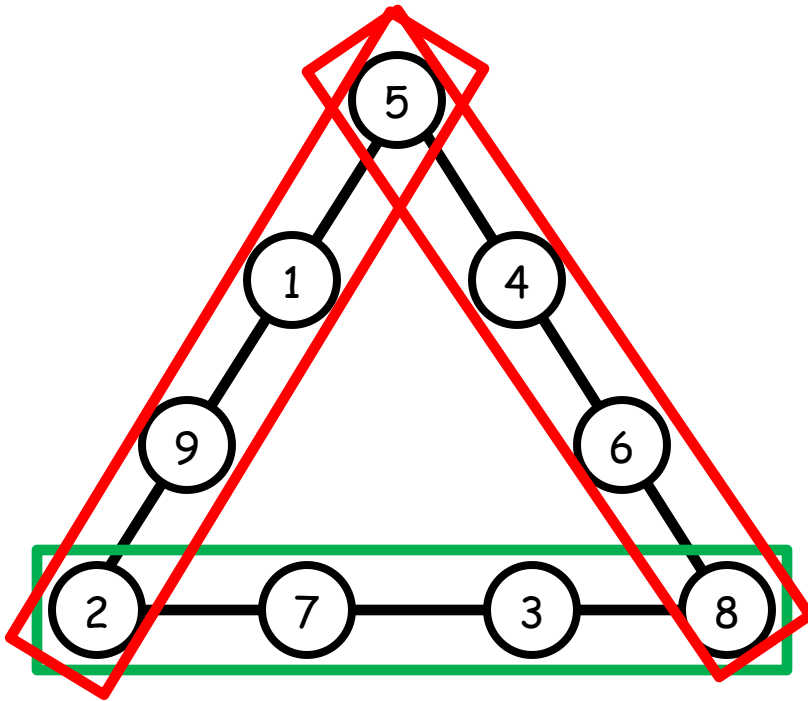
---



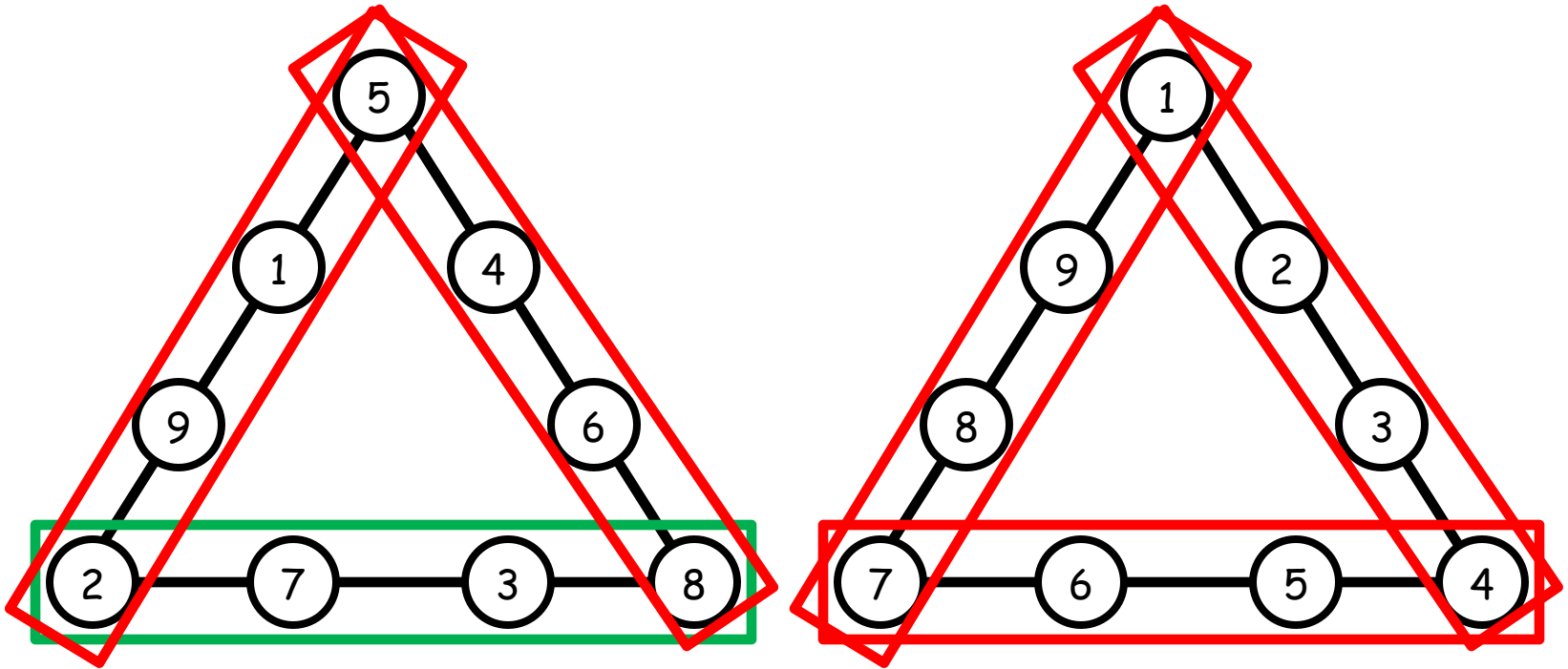
¿Cuál está más distante de la meta?  
¿Cómo lo sabe?

# Búsqueda informada

---

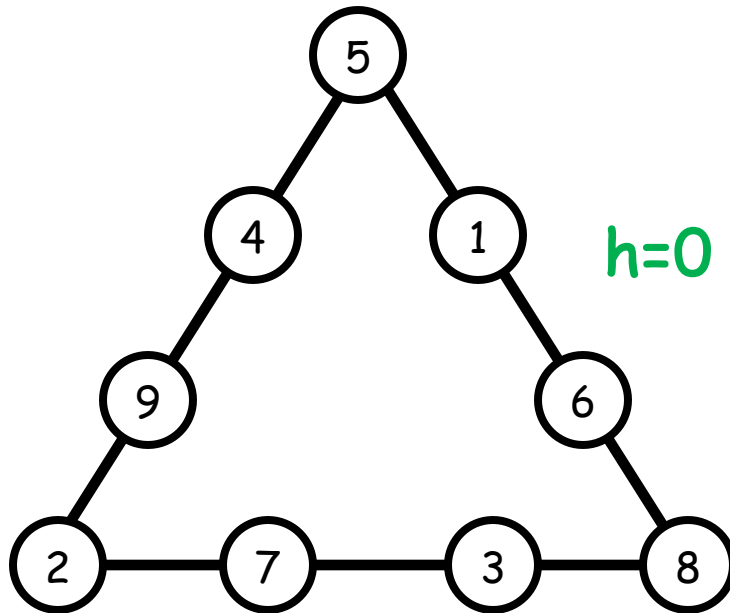
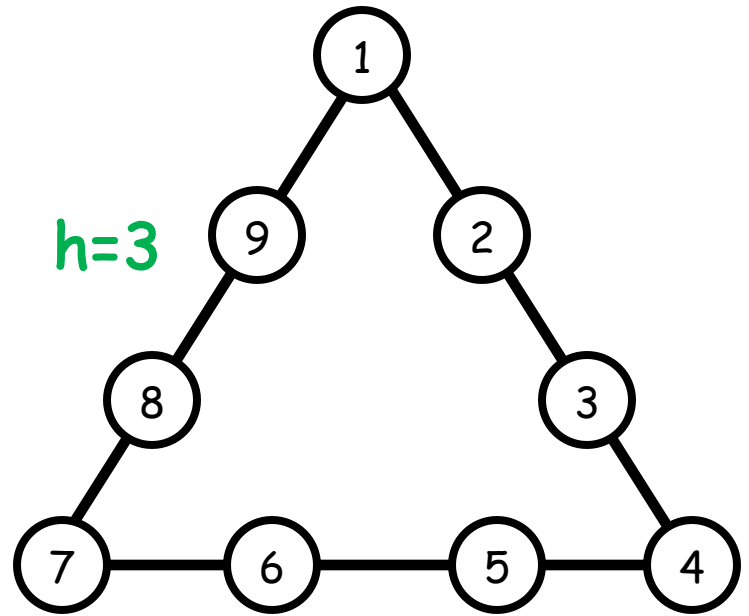
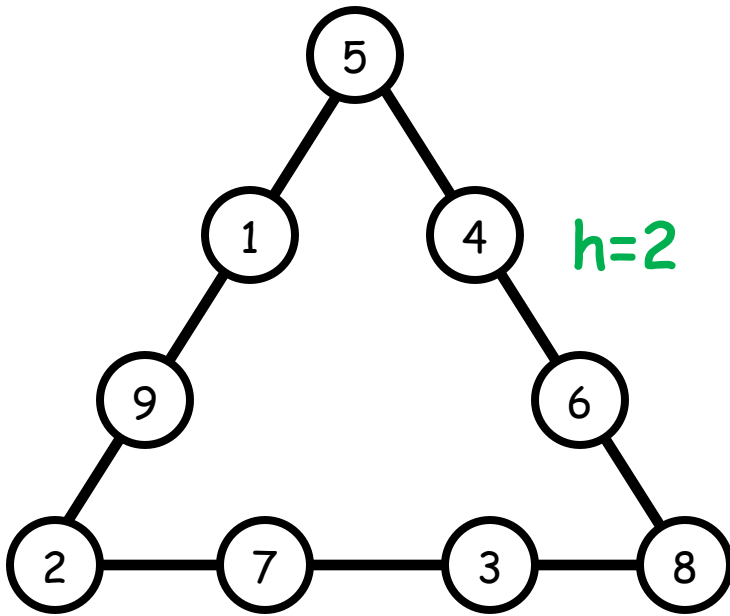


# Búsqueda informada



$h(n)$ : cantidad de lados que no suman 20





# Búsqueda informada

---

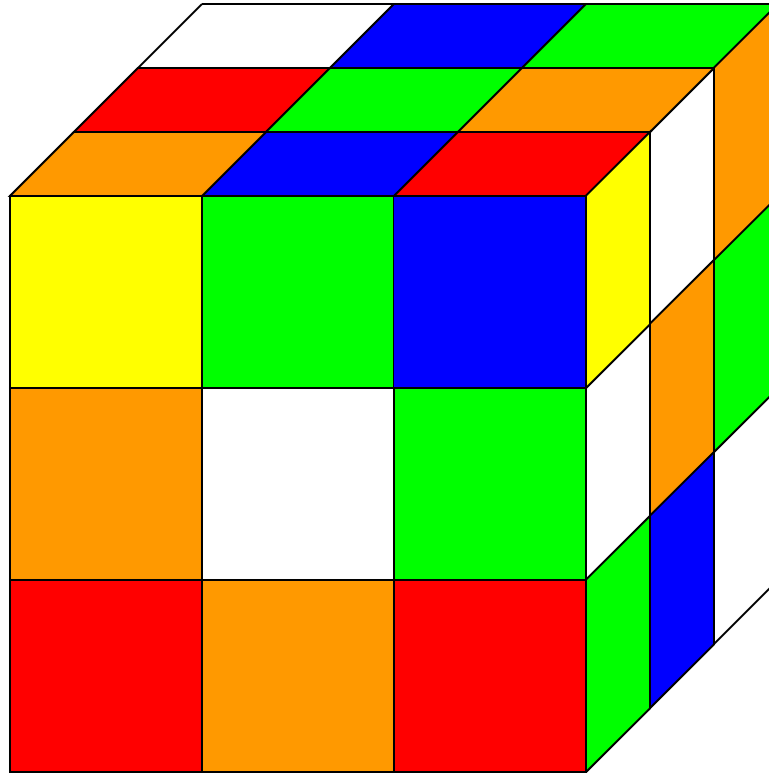
## Heurística

- La utilización de heurísticas en los métodos de búsqueda permiten reducir el espacio de estados a expandir en la construcción del árbol

# Búsqueda informada

---

## Cubo de Rubik



¿Cuál podría ser una heurística?

Entre diferentes estados del ambiente, ¿cómo saber cuál está más distante de la meta?

# Búsqueda informada

---

## Cubo de Rubik



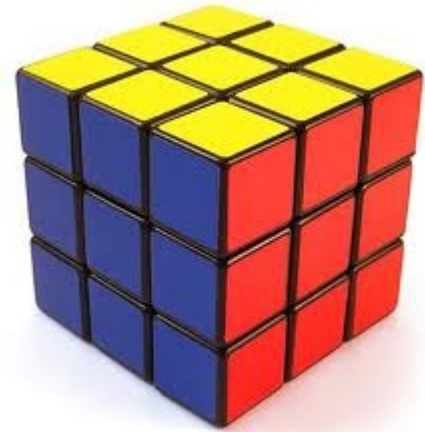
¿Cuál podría ser una heurística?

Entre diferentes estados del ambiente, ¿cómo saber cuál está más distante de la meta?

# Búsqueda informada

---

## Cubo de Rubik



¿Cuál podría ser una heurística?

Entre diferentes estados del ambiente, ¿cómo saber cuál está más distante de la meta?

# Búsqueda informada

---

## Estrategias de búsqueda informada

- Avara
- $A^*$

# Búsqueda informada

---

## Búsqueda Avara

- Expandir el nodo con menor  $h(n)$ , esto es, aquel que parece estar más cerca de la meta, sin importar a qué profundidad esté

# Búsqueda informada

---

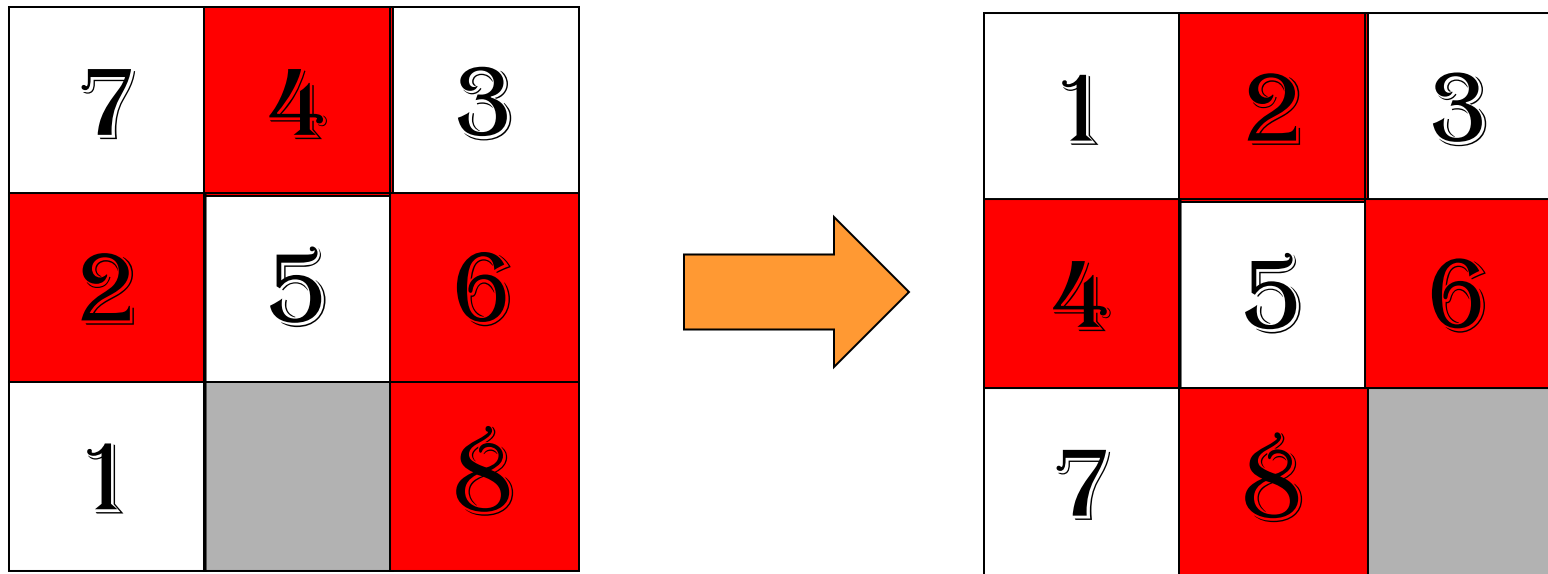
## Búsqueda Avara

- Expandir el nodo con menor  $h(n)$ , esto es, aquel que parece estar más cerca de la meta, sin importar a qué profundidad esté
- Intenta reducir al mínimo el costo estimado para lograr una meta



# Búsqueda informada

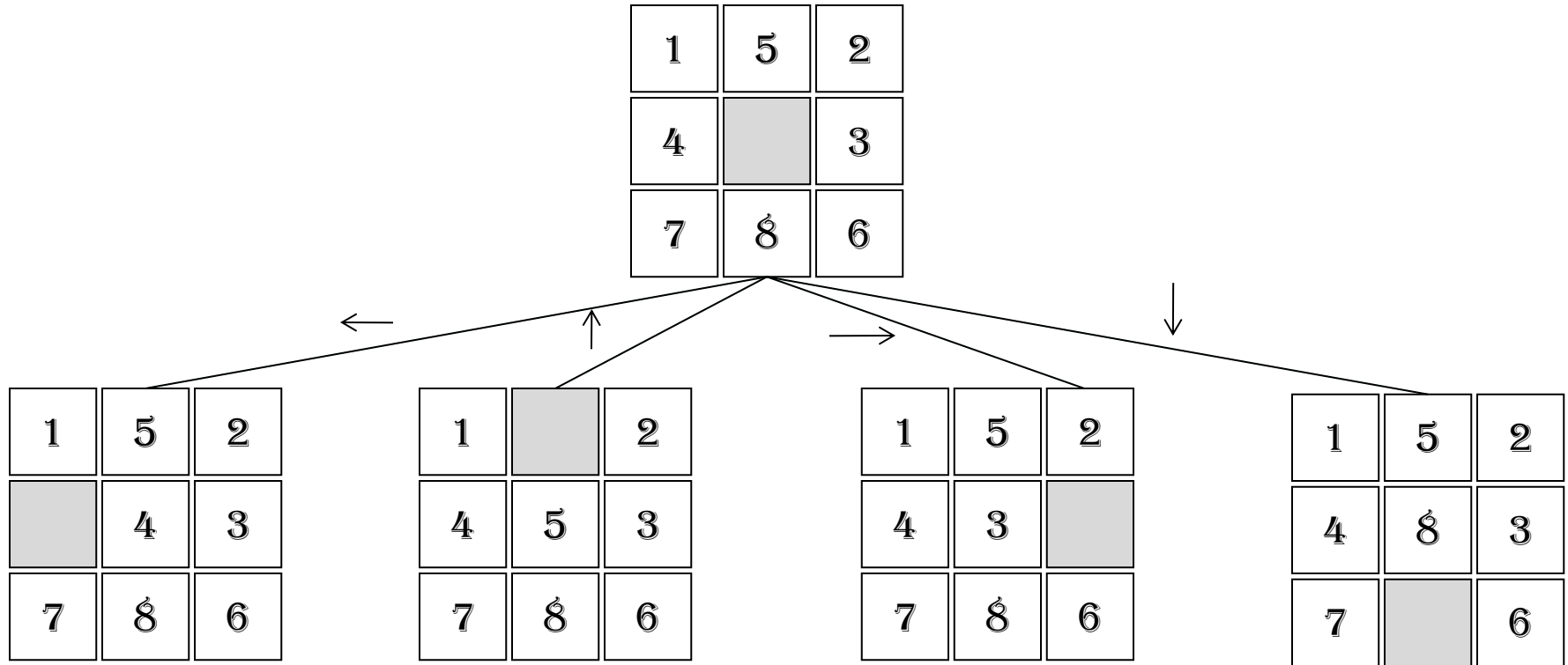
## 8-puzzle



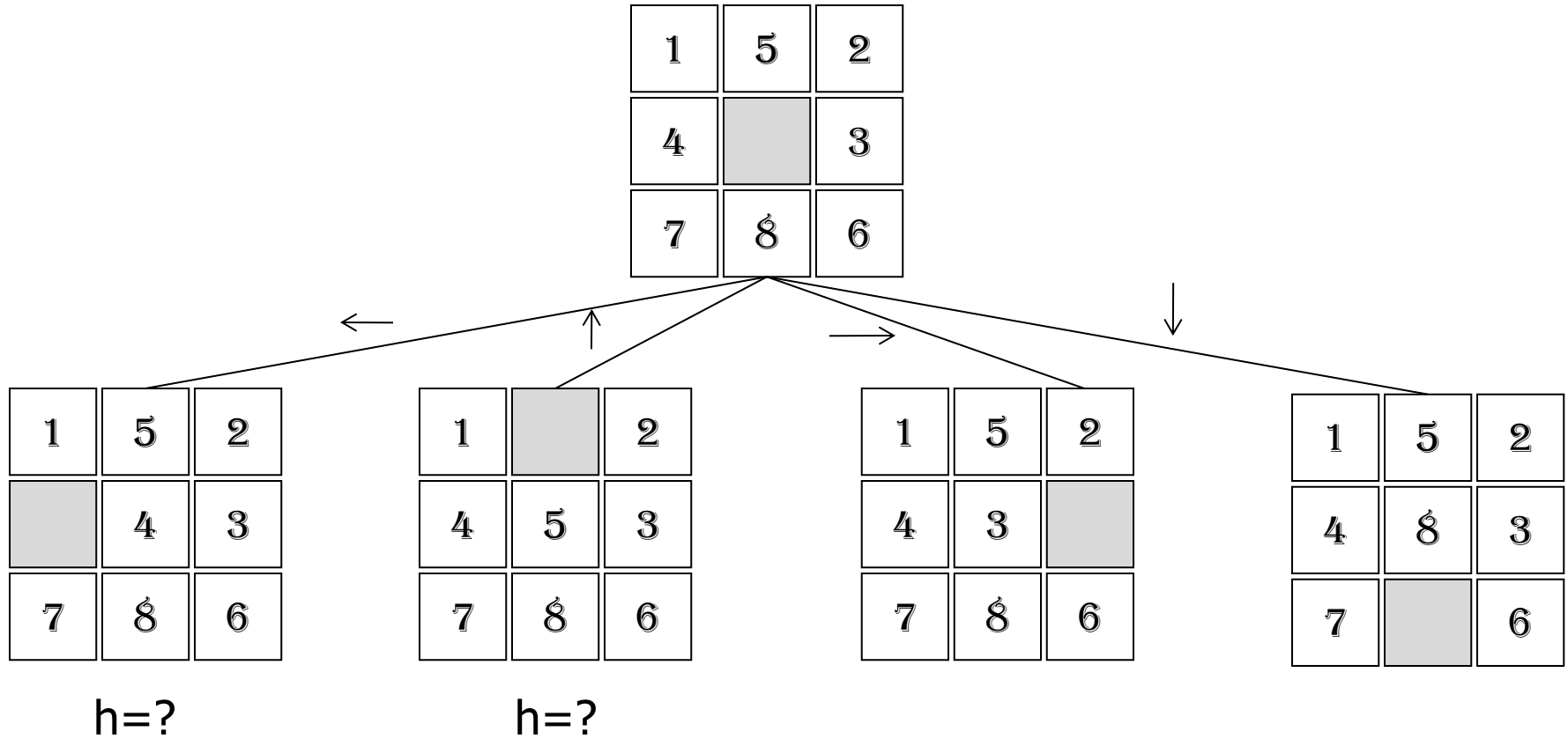
$h(n)$ : cantidad de piezas ubicadas en el lugar incorrecto (teniendo en cuenta la placa vacía)

# Búsqueda informada

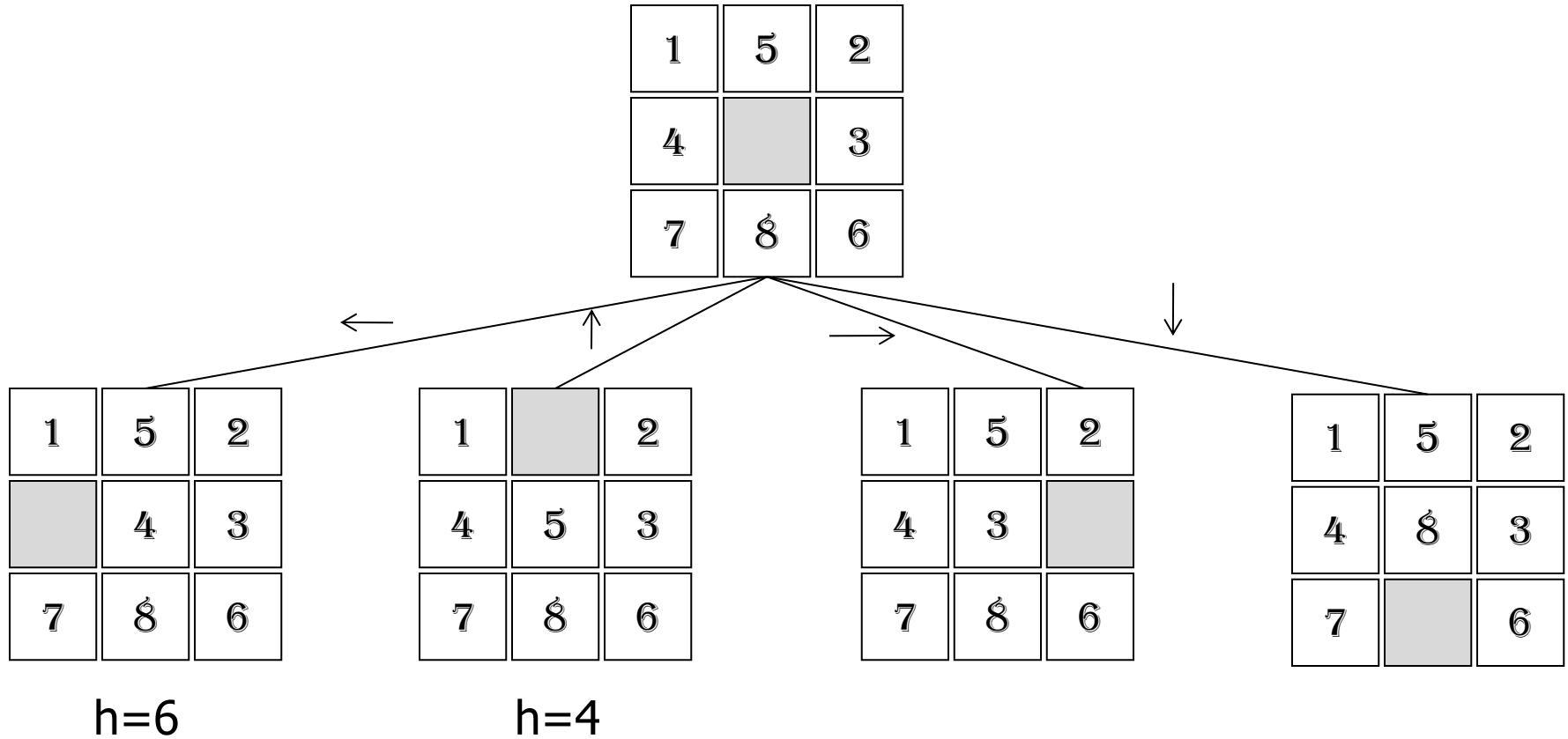
---



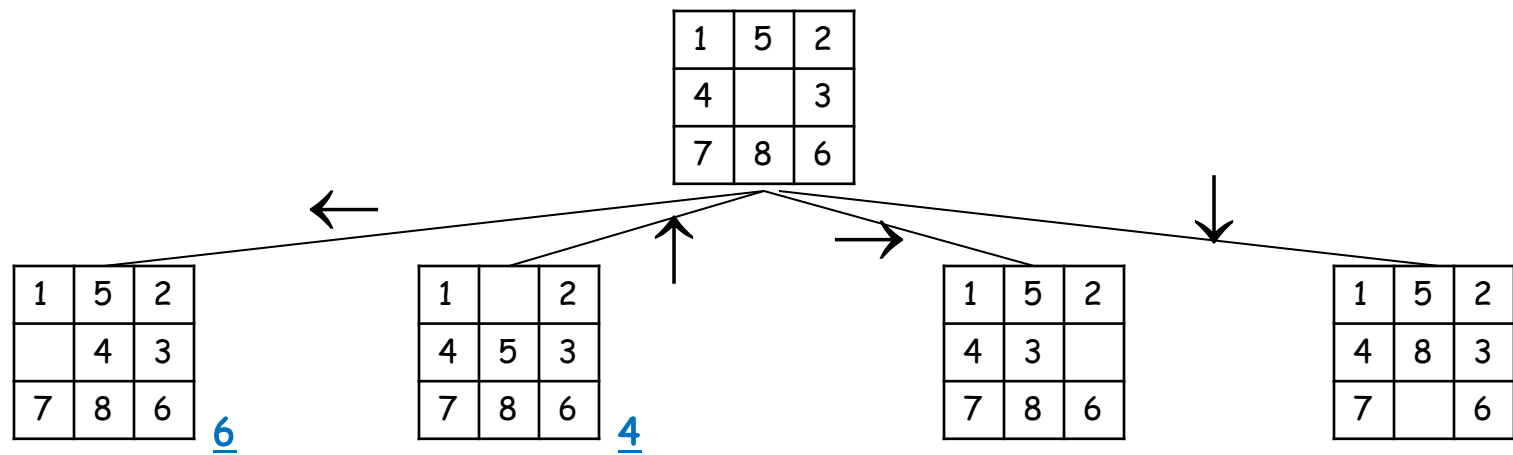
# Búsqueda informada



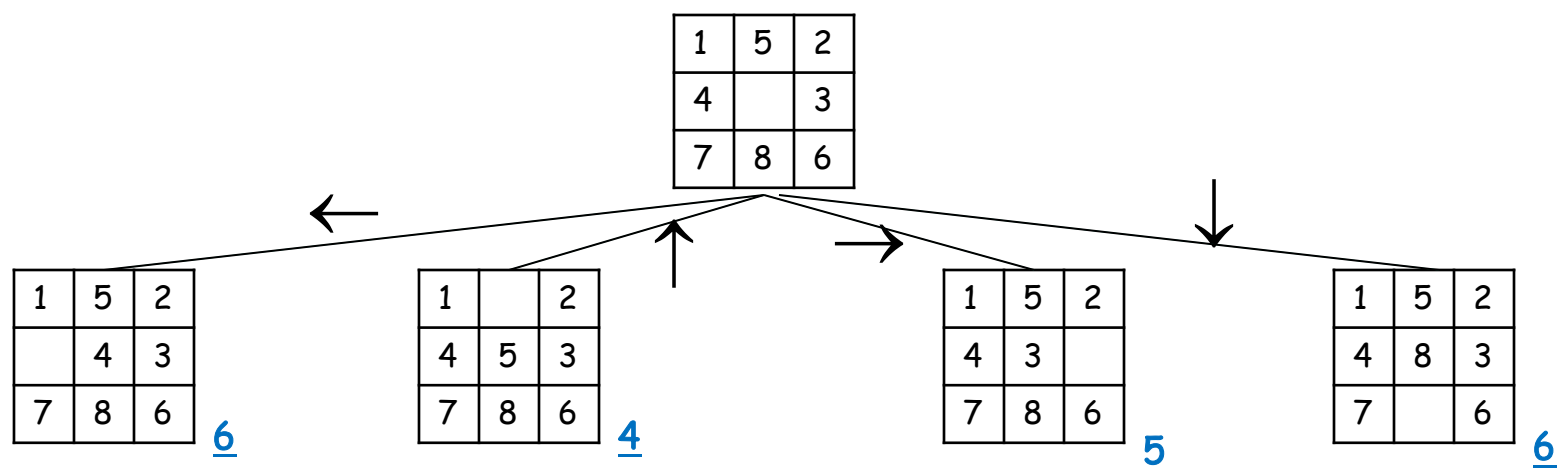
# Búsqueda informada

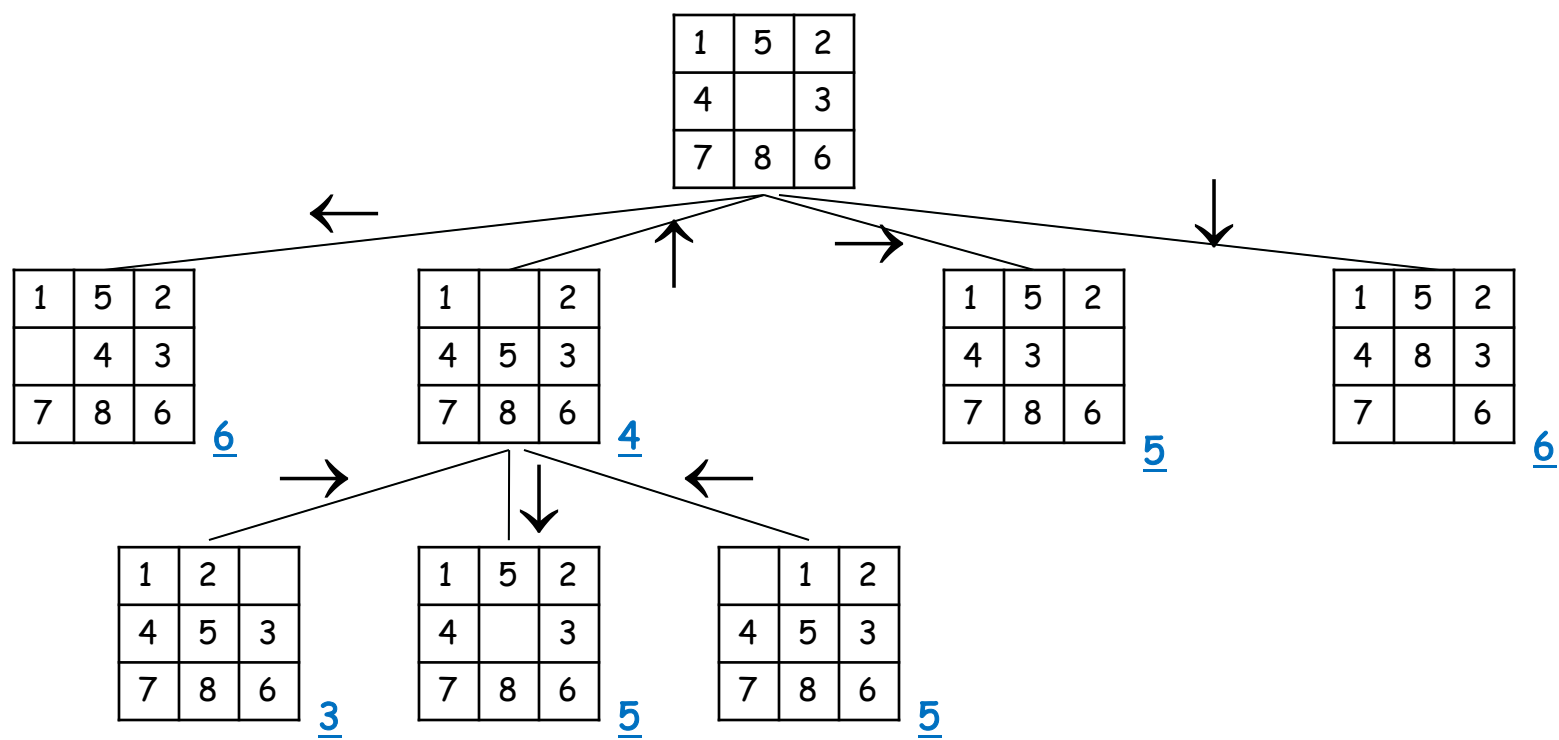


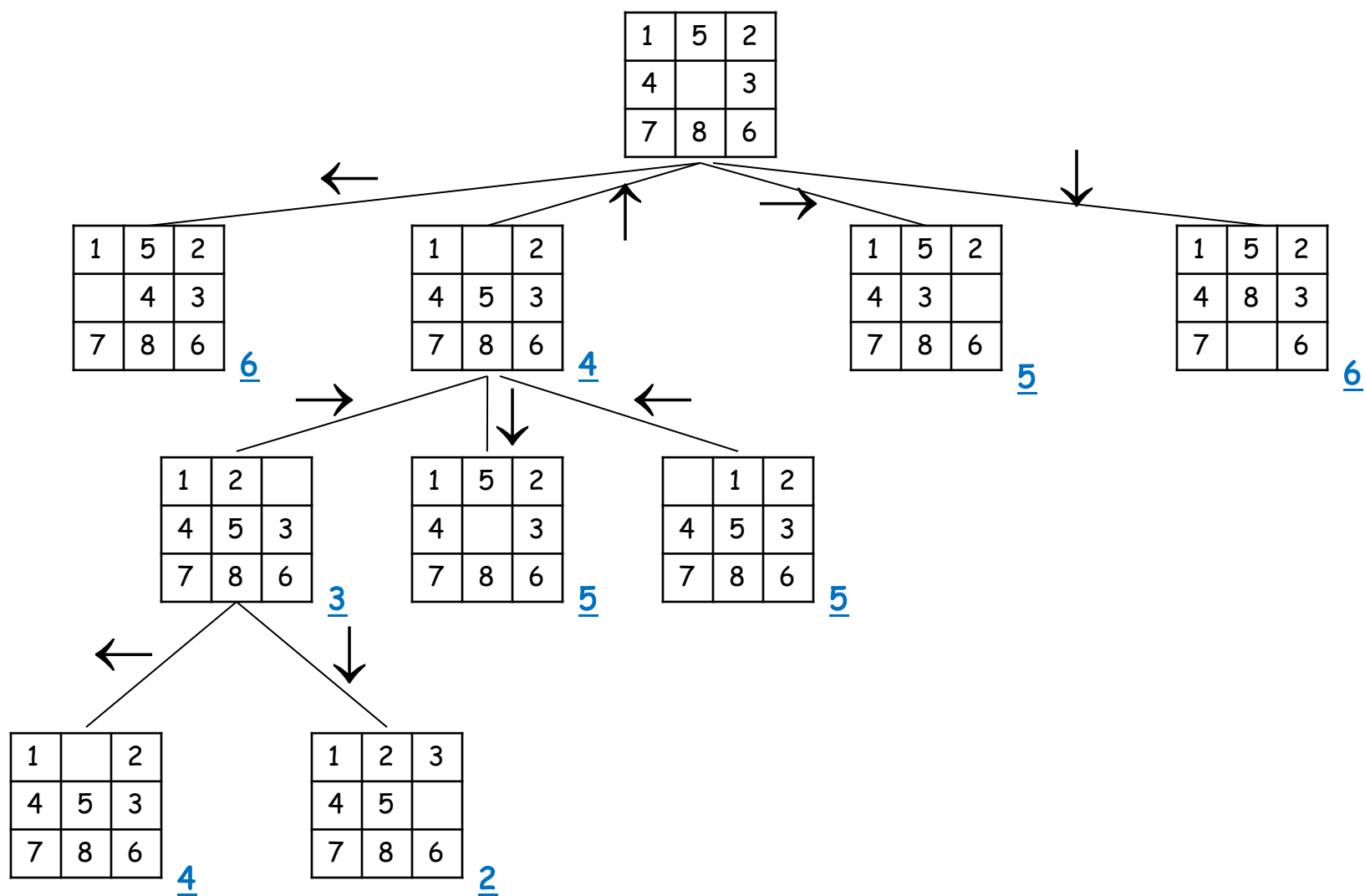
Complete el árbol de búsqueda  
No evite devolverse



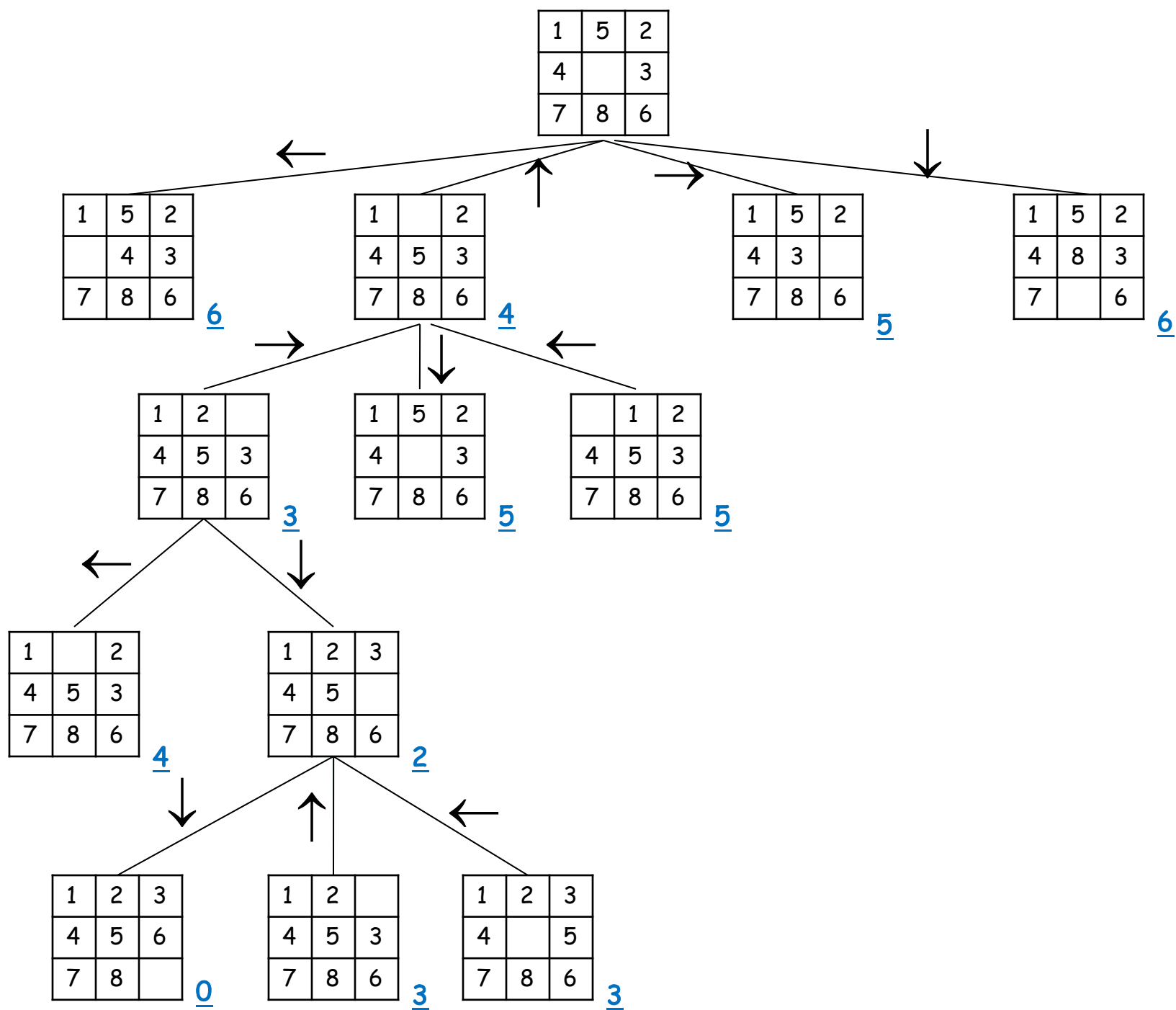
Complete el árbol de búsqueda  
No evite devolverse

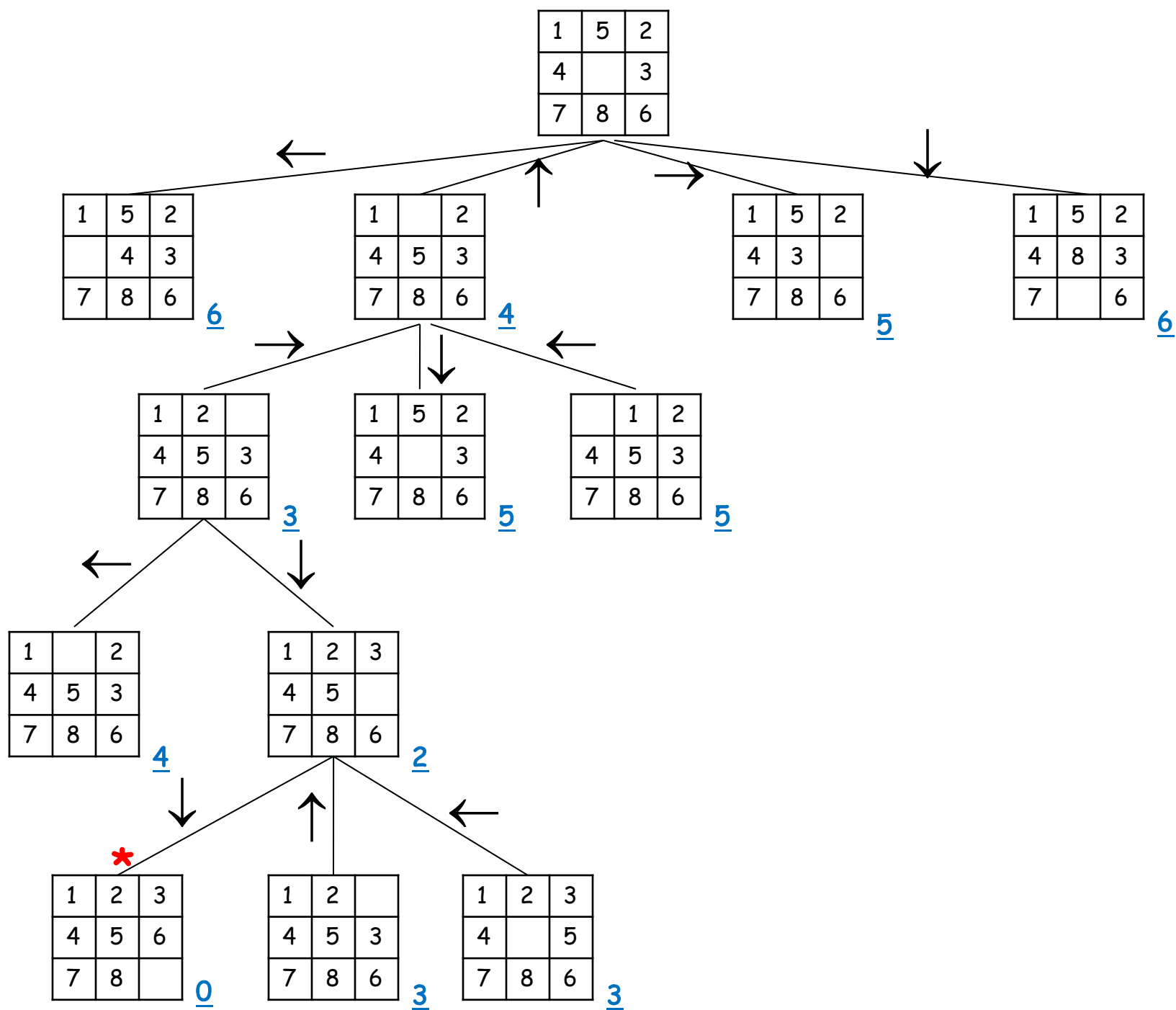




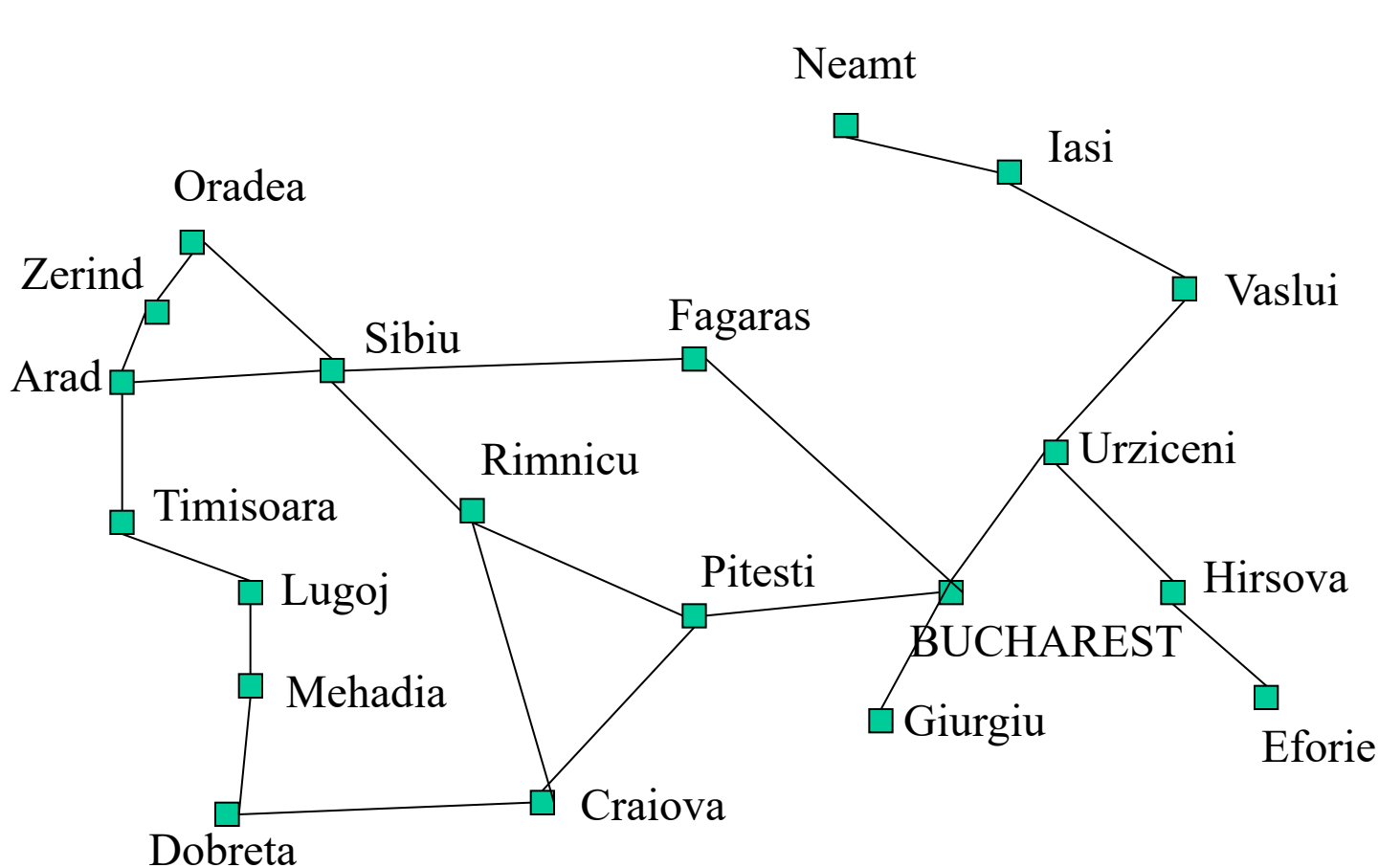








# Búsqueda informada



Distancia en  
línea recta a  
Bucarest:

Arad	366
Bucarest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

$h(n)$ : distancia en línea recta entre  $n$  y la ubicación de la meta

No evite devolverse

# Búsqueda informada

---

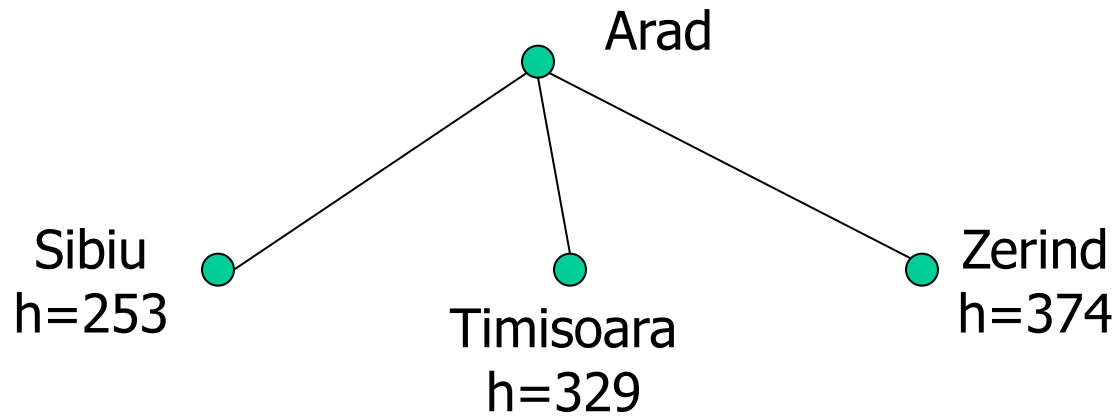
No evite devolverse



Arad

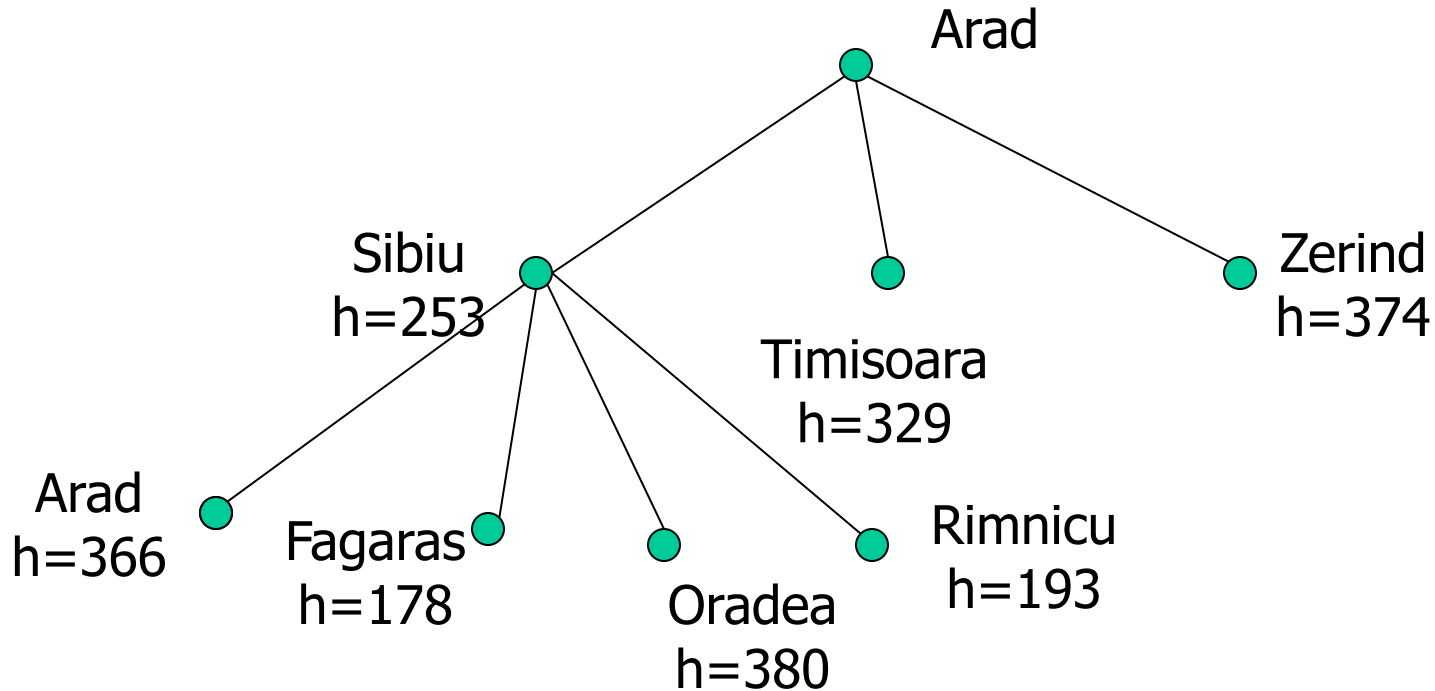
# Búsqueda informada

---



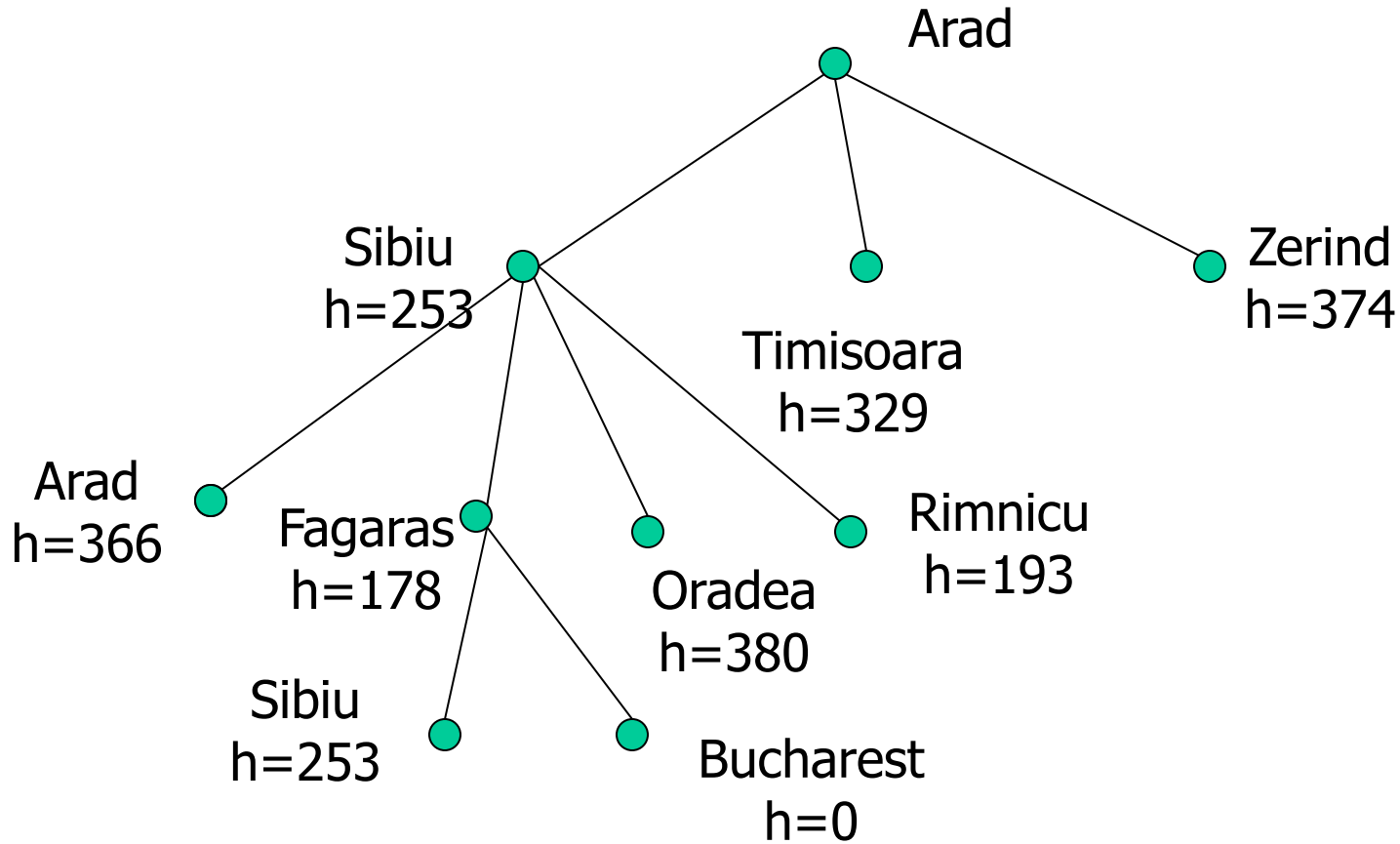
# Búsqueda informada

---

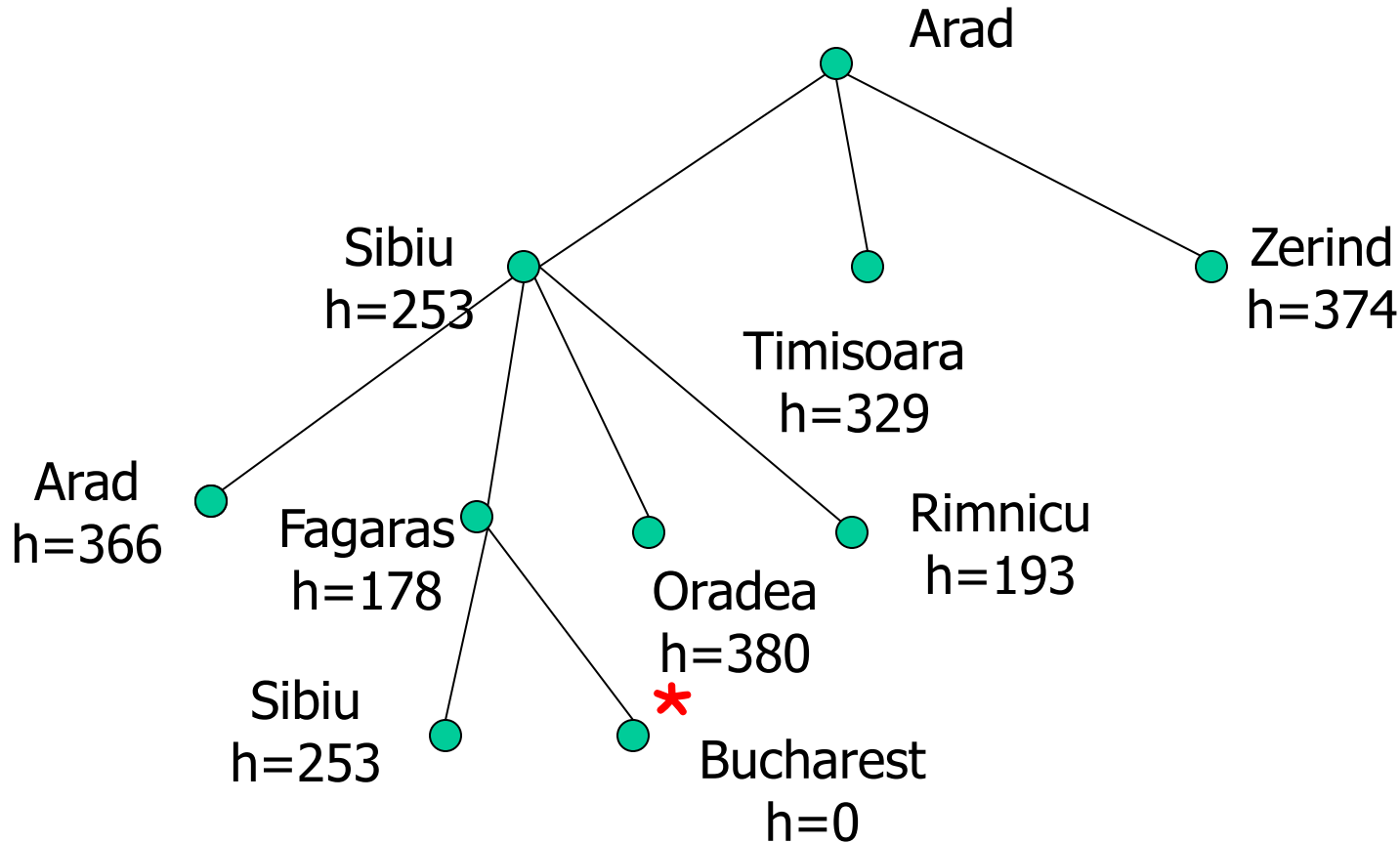


# Búsqueda informada

---



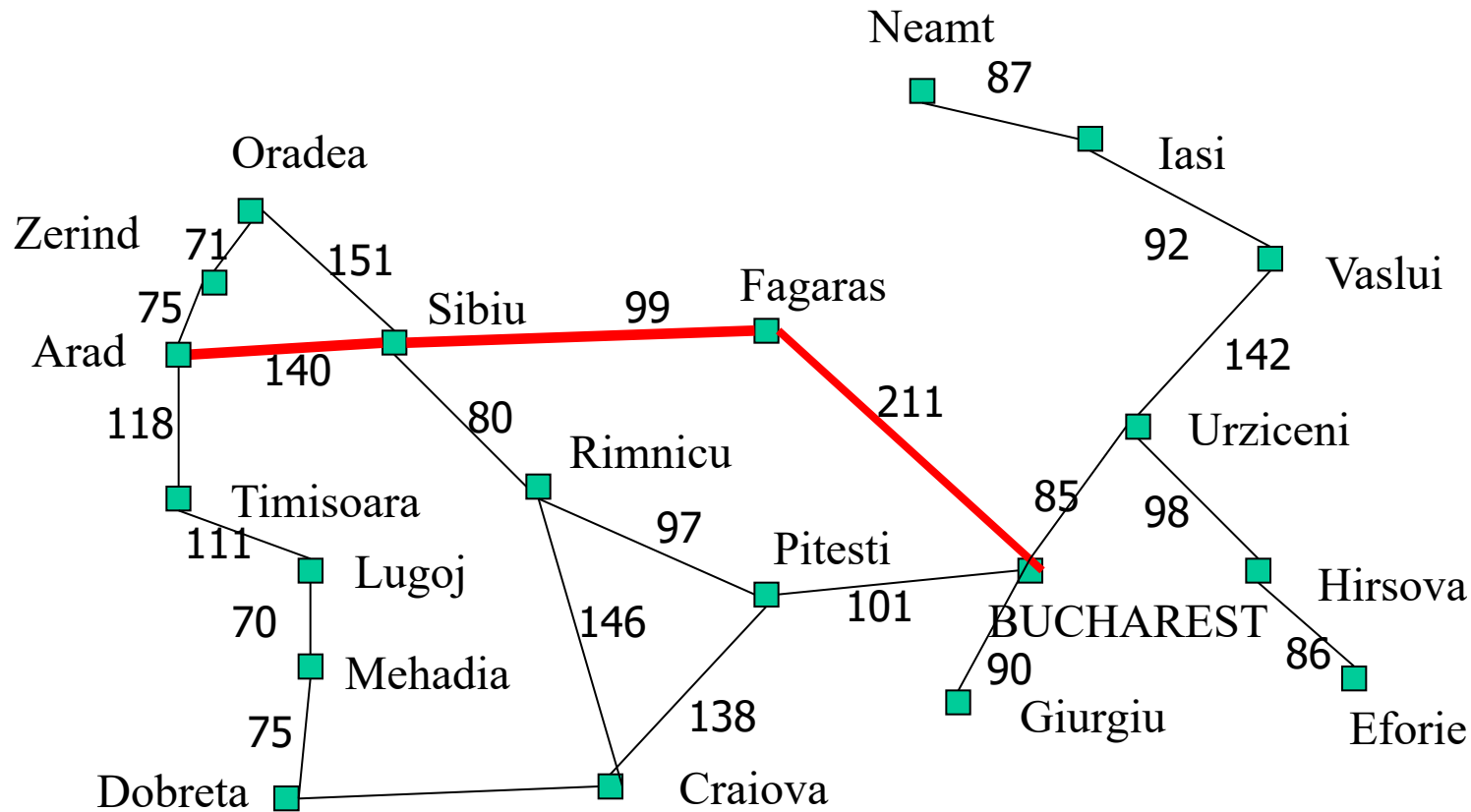
# Búsqueda informada



El camino obtenido es  
*Arad* → *Sibiu* → *Fagaras* → *Bucarest*

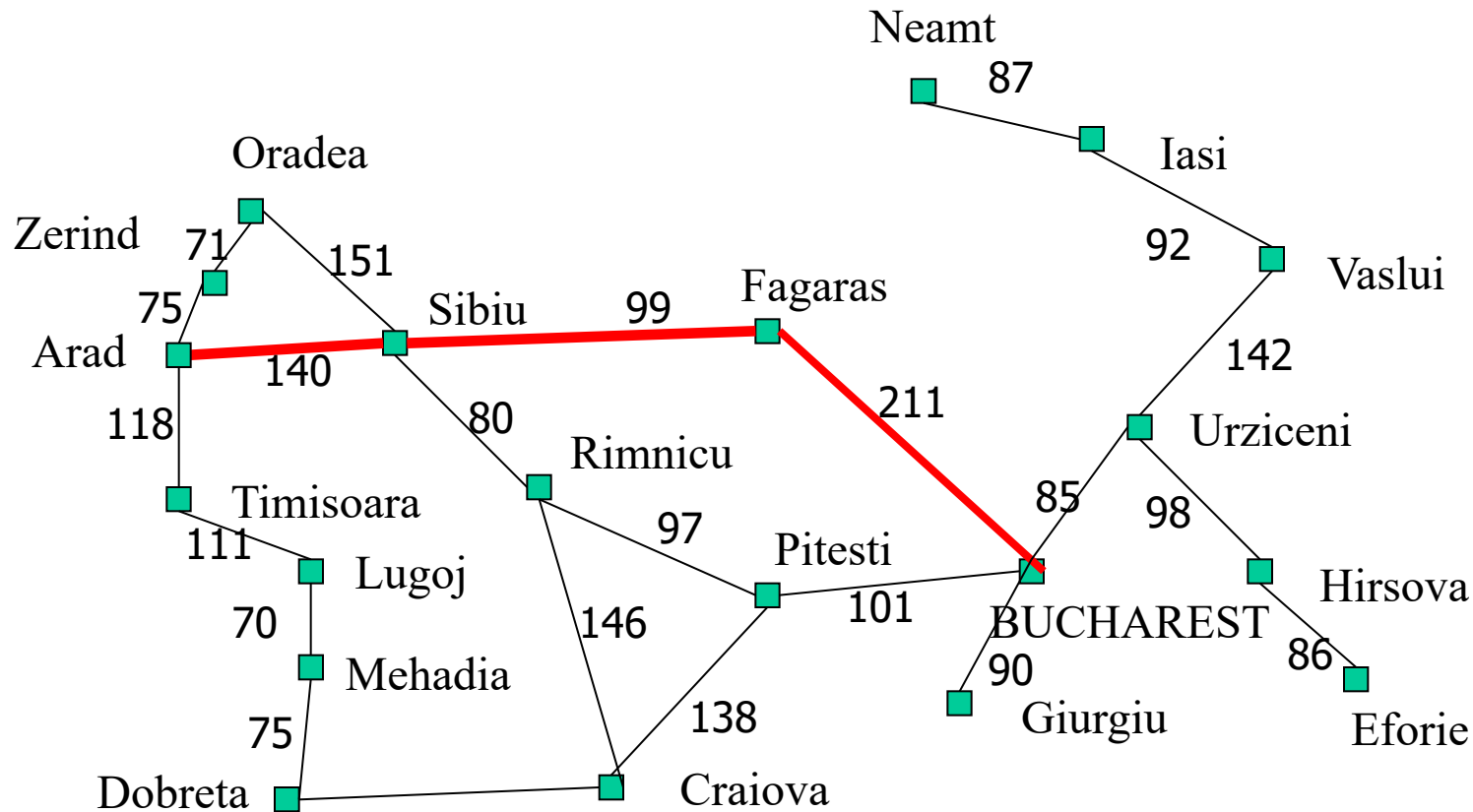


# Búsqueda informada



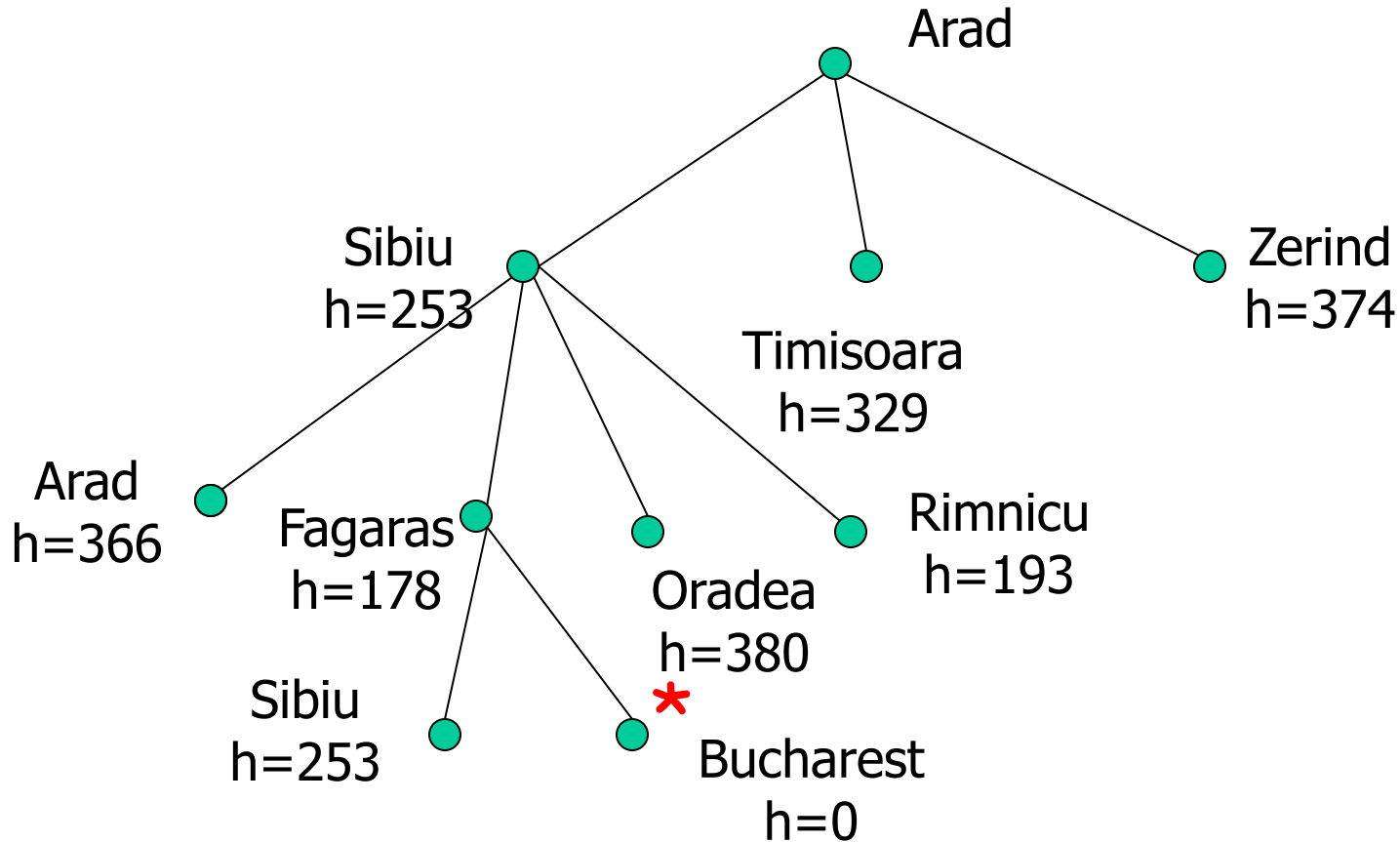
El camino obtenido es  
*Arad* → *Sibiu* → *Fagaras* → *Bucarest*

# Búsqueda informada



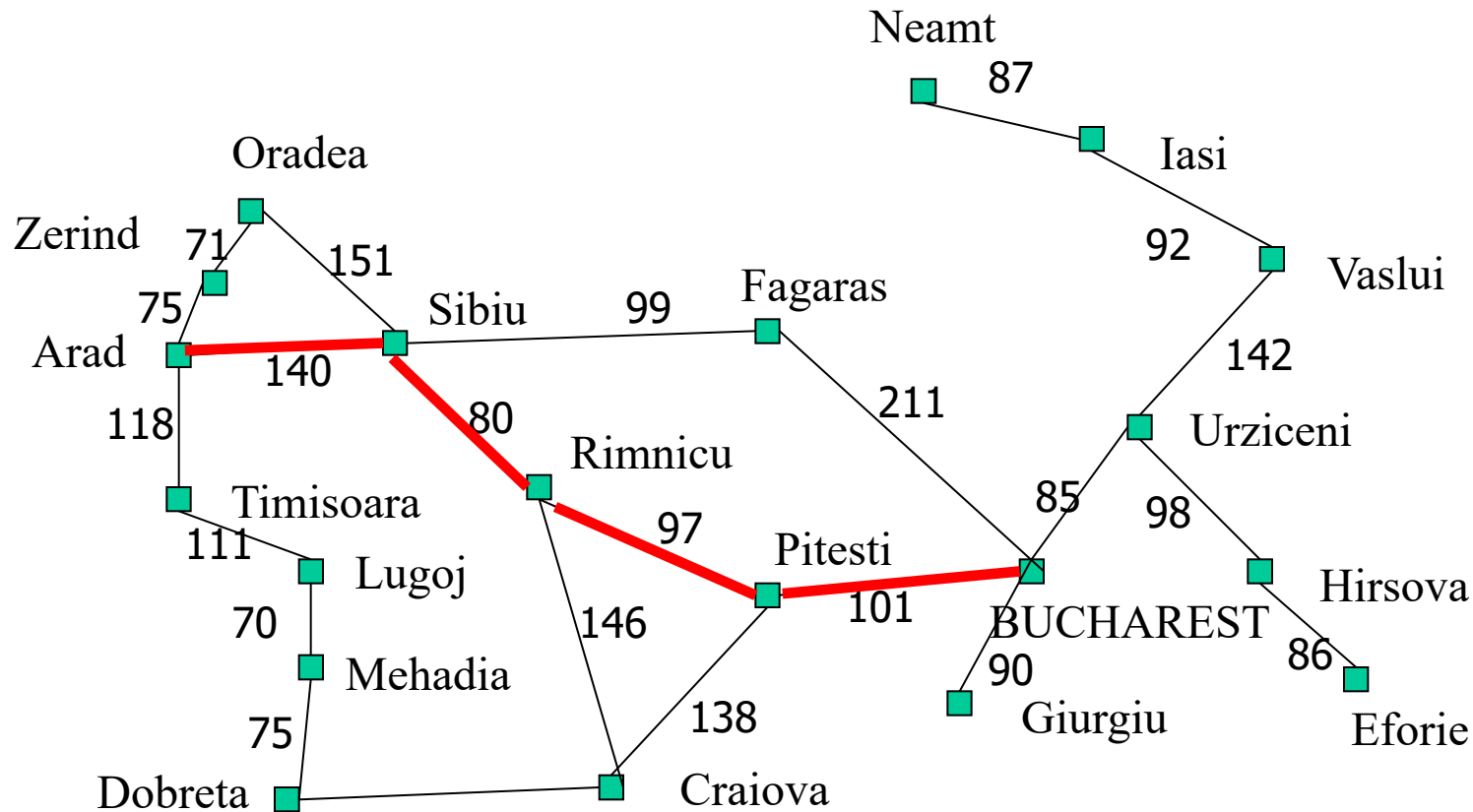
El costo real de la solución  
*Arad* → *Sibiu* → *Fagaras* → *Bucarest* es 450

# Búsqueda informada



¿Es ésta la solución óptima?

# Búsqueda informada



El camino óptimo es  
*Arad* → *Sibiu* → *Rimnicu* → *Pitesti* → *Bucarest*  
porque tiene costo **418**

# Búsqueda informada

---

- La búsqueda avara no es óptima

# Búsqueda informada

---

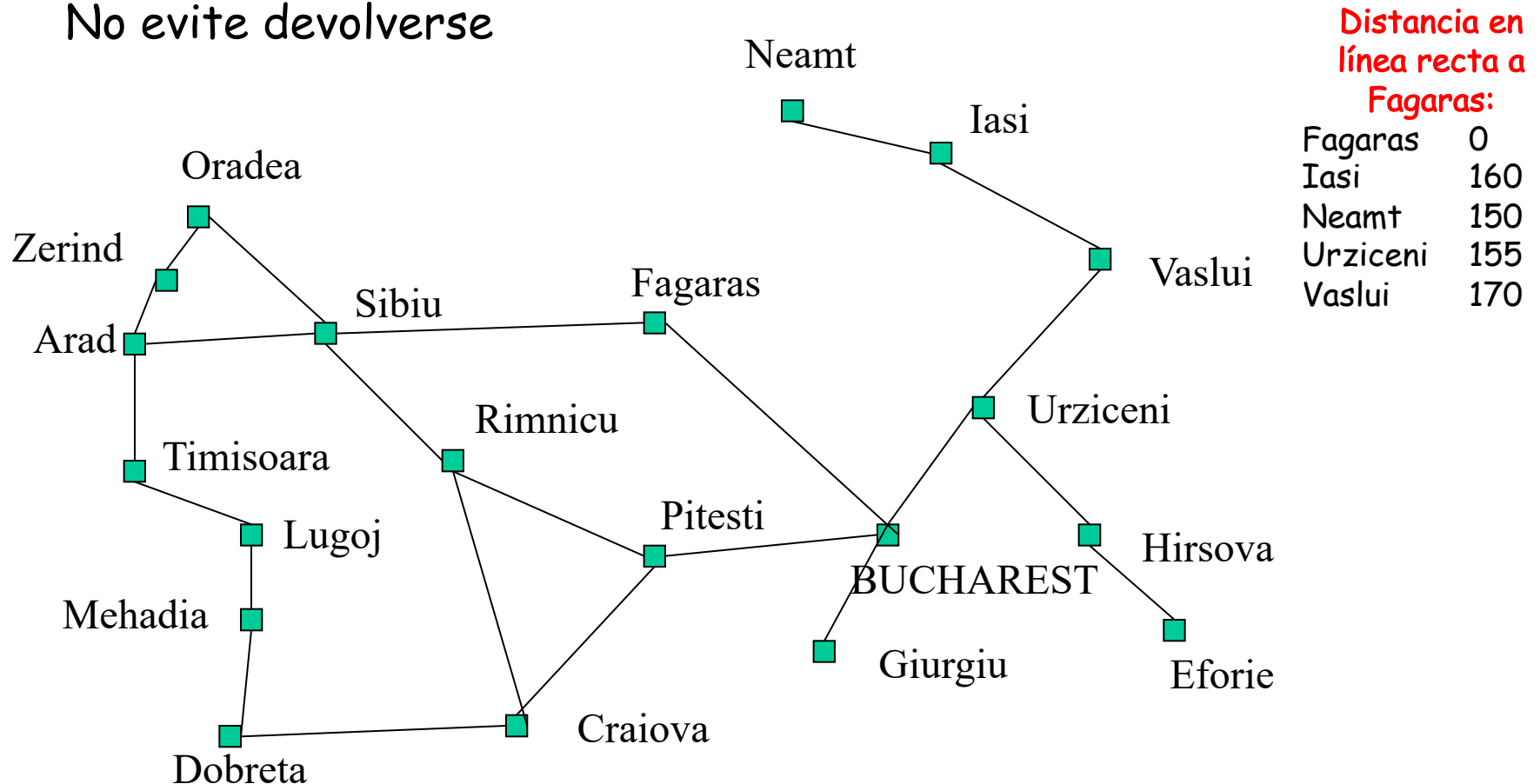
- Construya el árbol para ir de **Iasi** a **Fagaras**

No evite devolverse

# Búsqueda informada

- Construya el árbol para ir de Iasi a Fagaras

No evite devolverse



# Búsqueda informada

---

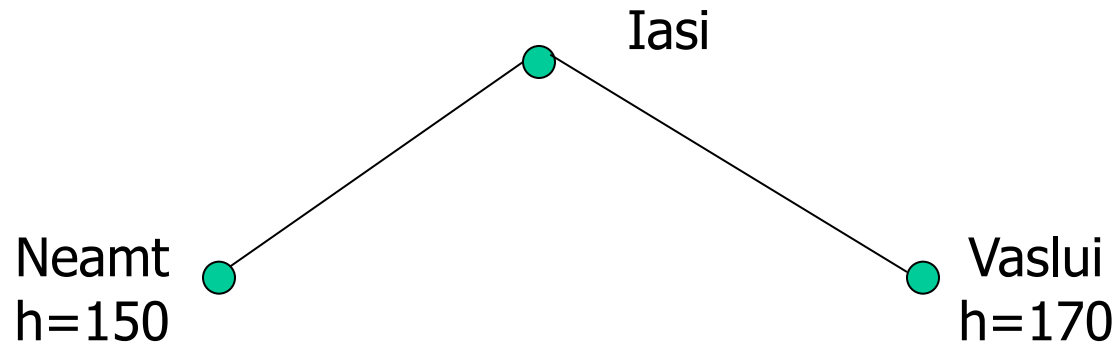


Iasi



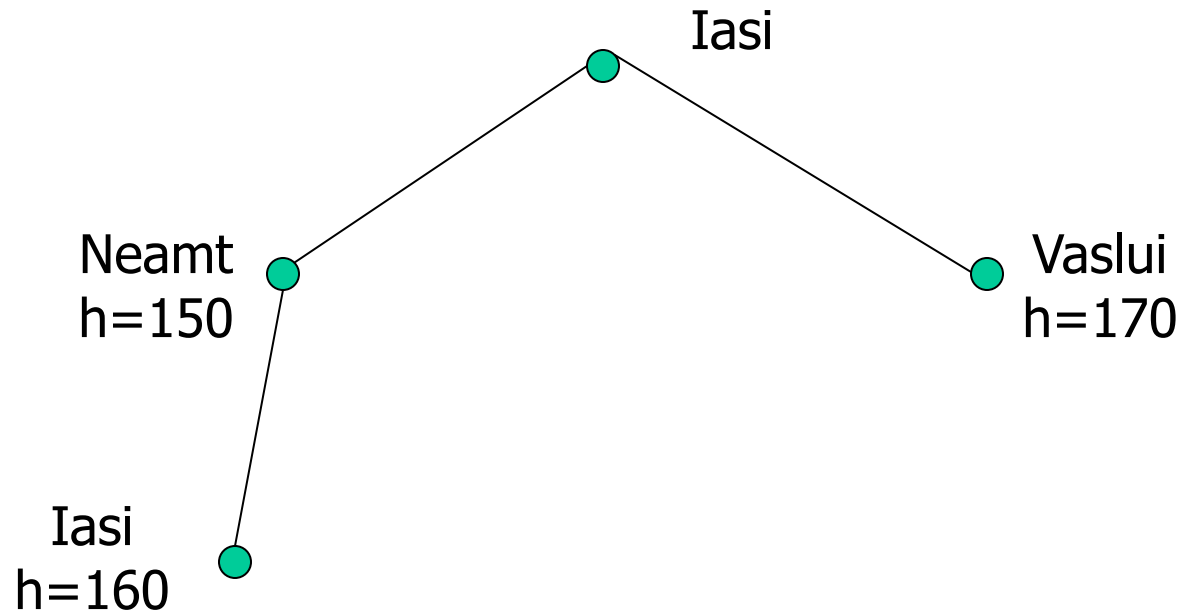
# Búsqueda informada

---



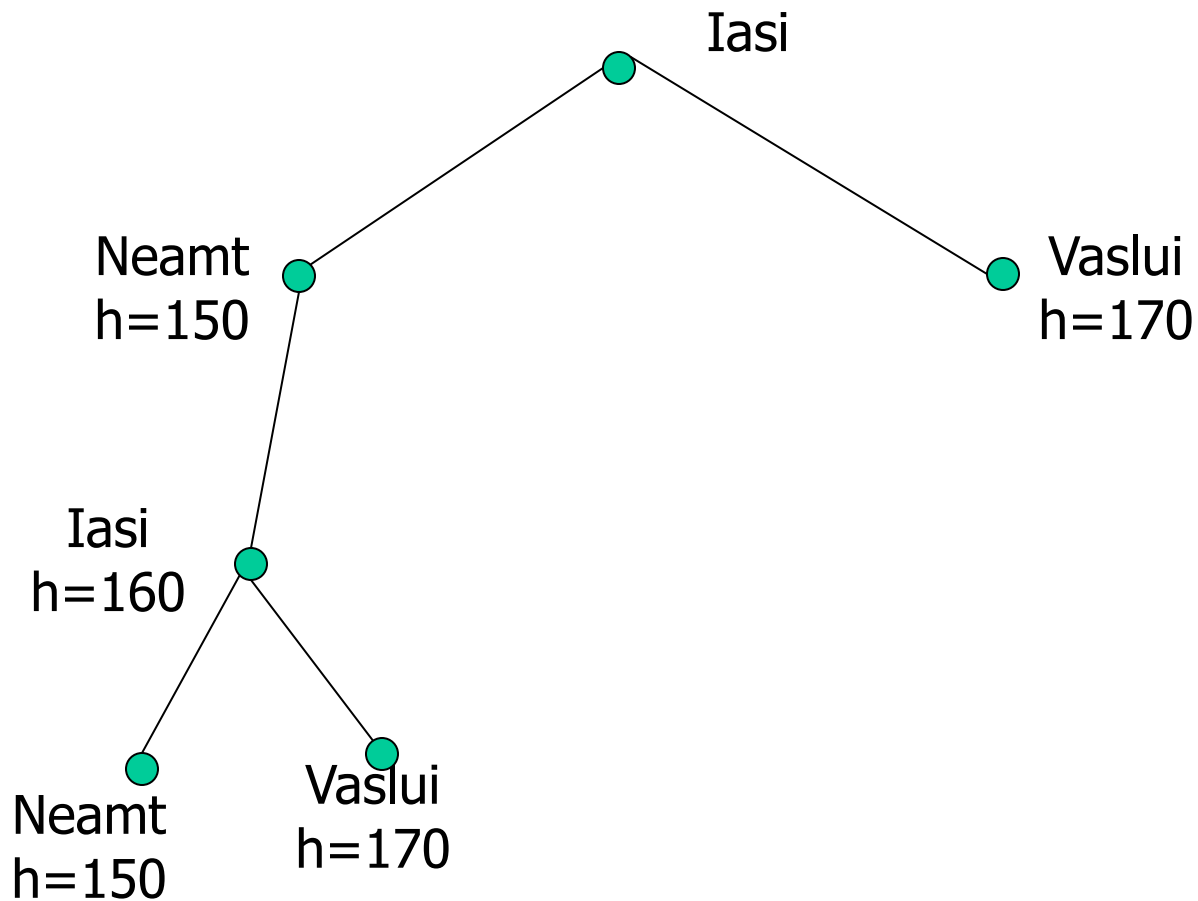
# Búsqueda informada

---



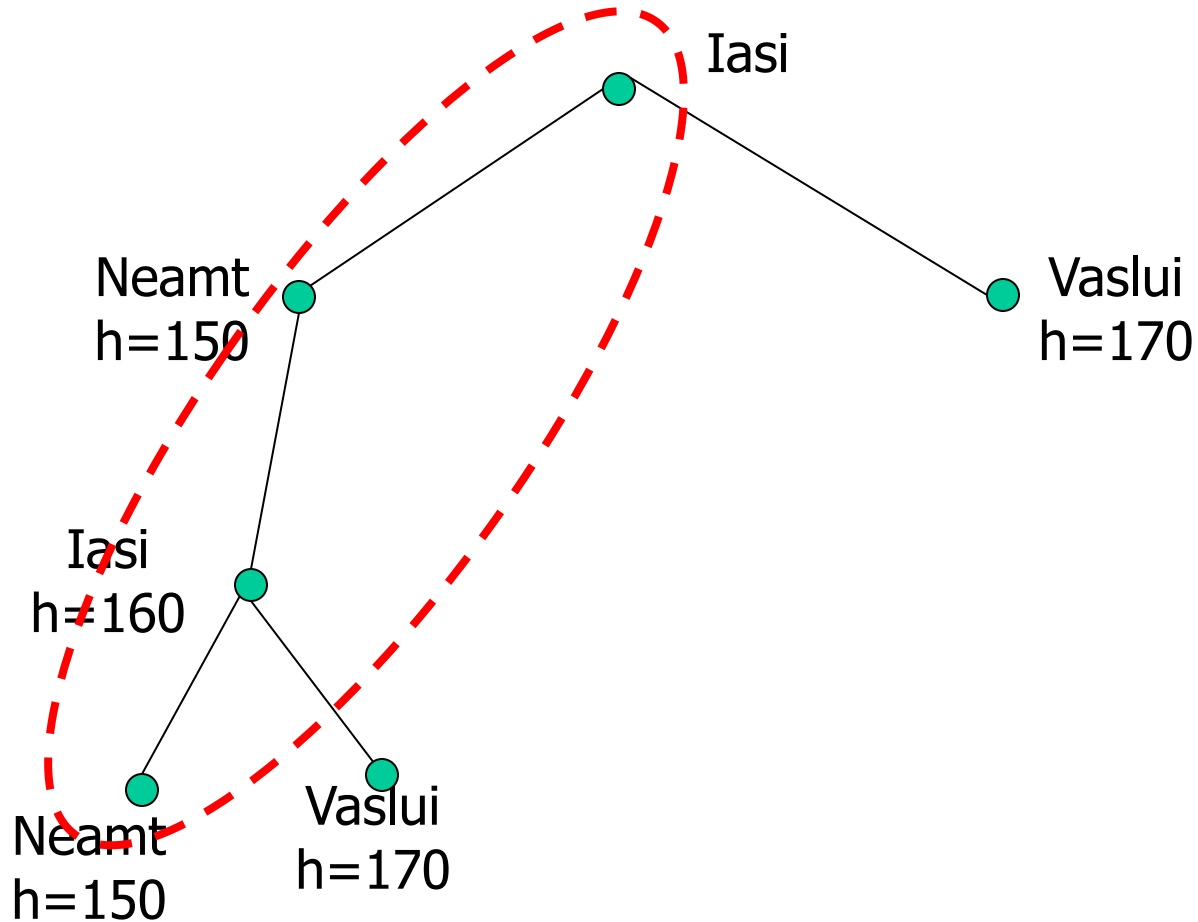
# Búsqueda informada

---



# Búsqueda informada

---



Se queda en un ciclo

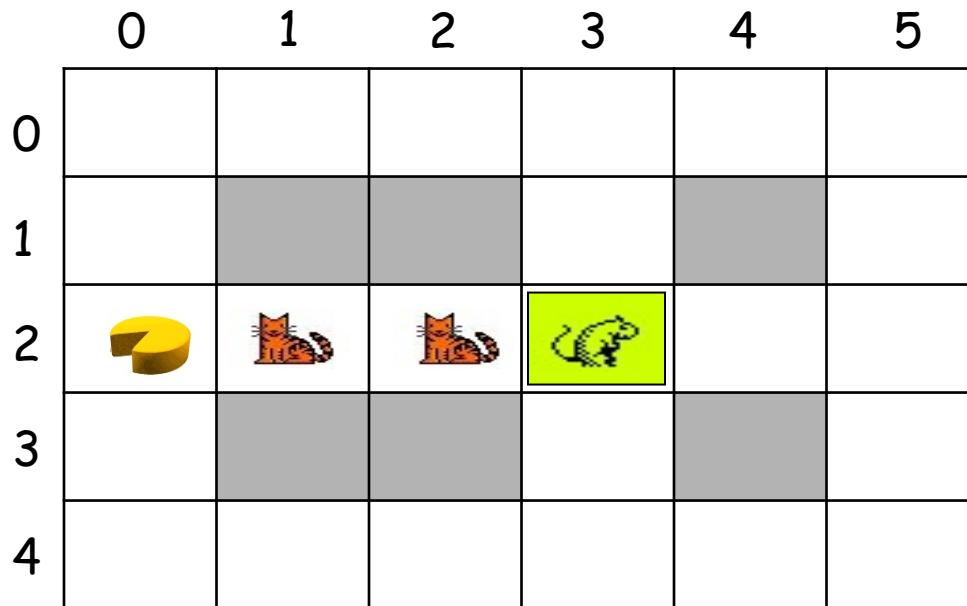
# Búsqueda informada

---

- La búsqueda avara **no es completa**

# Búsqueda informada

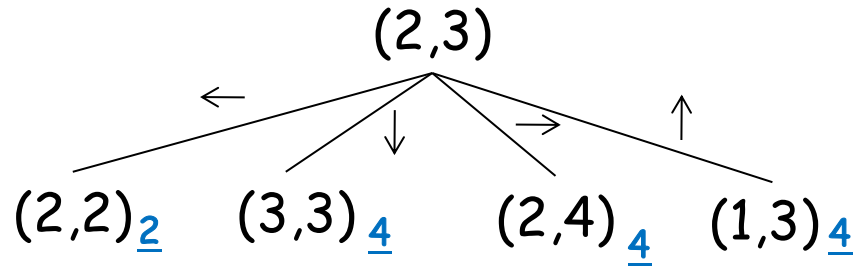
## El ratón inteligente







$h(n)$ : suma de las distancias horizontales y verticales para alcanzar el queso. Distancia de Manhattan

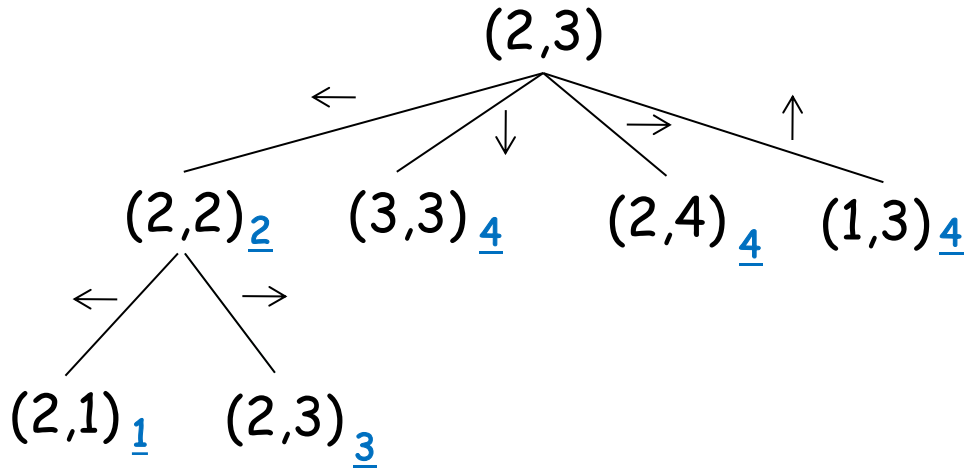
No evite devolverse





# Búsqueda informada



	0	1	2	3	4	5
0						
1						
2						
3						
4						

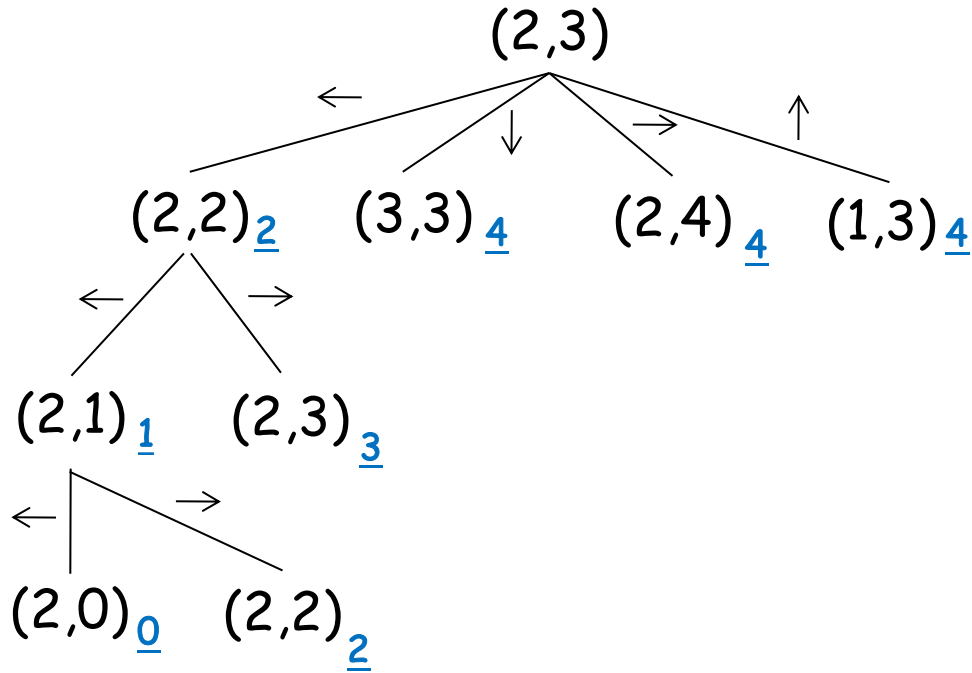
# Búsqueda informada







	0	1	2	3	4	5
0						
1						
2						
3						
4						

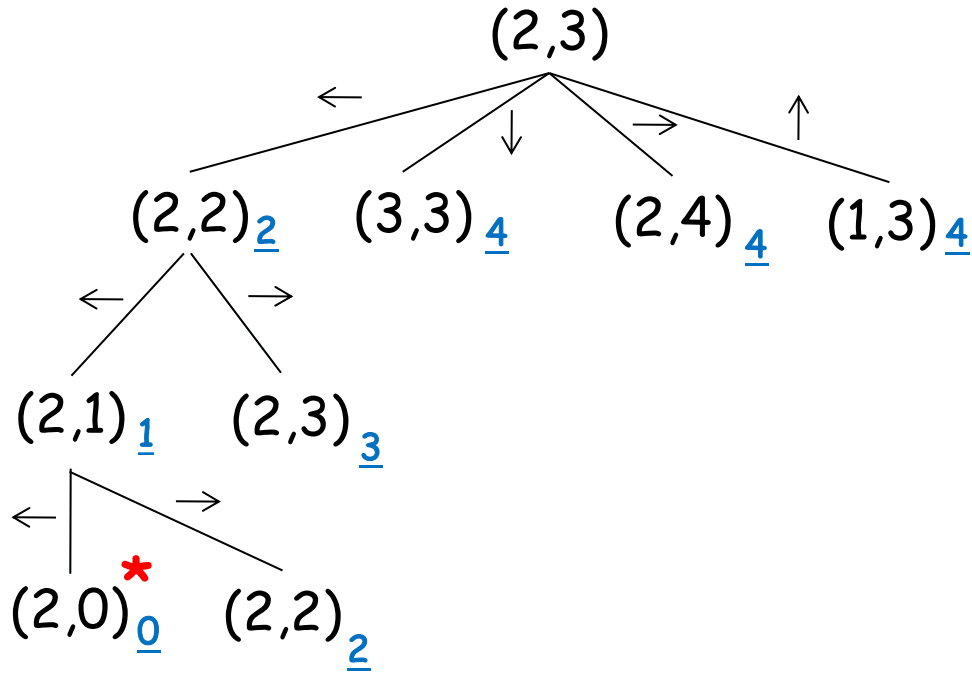






# Búsqueda informada



	0	1	2	3	4	5
0						
1						
2						
3						
4						

# Búsqueda informada

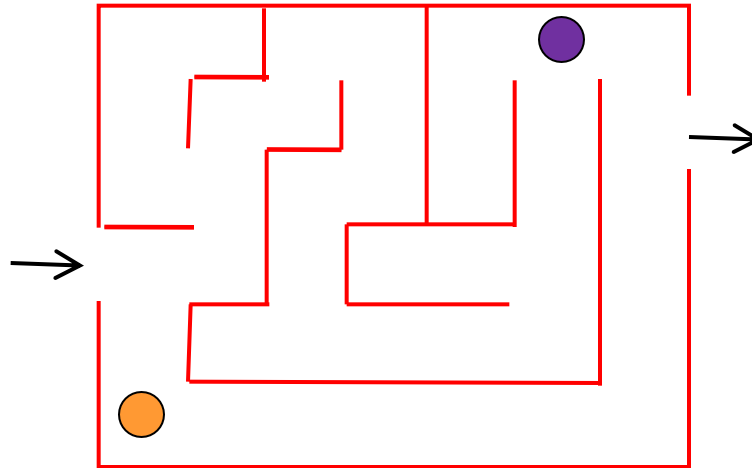


	0	1	2	3	4	5
0						
1						
2						
3						
4						

# Búsqueda informada

---

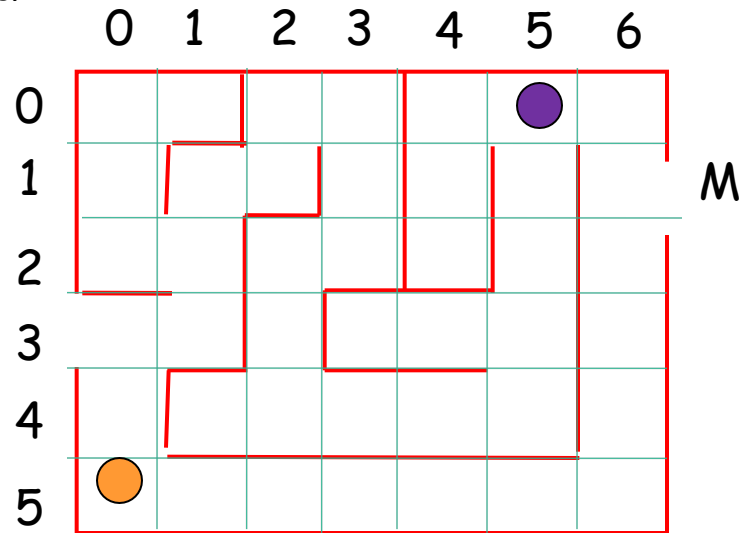
- Definir una heurística
- Aplicar búsqueda avara



¿Cuál está más distante de la meta?  
¿Cómo lo sabe?

# Búsqueda informada

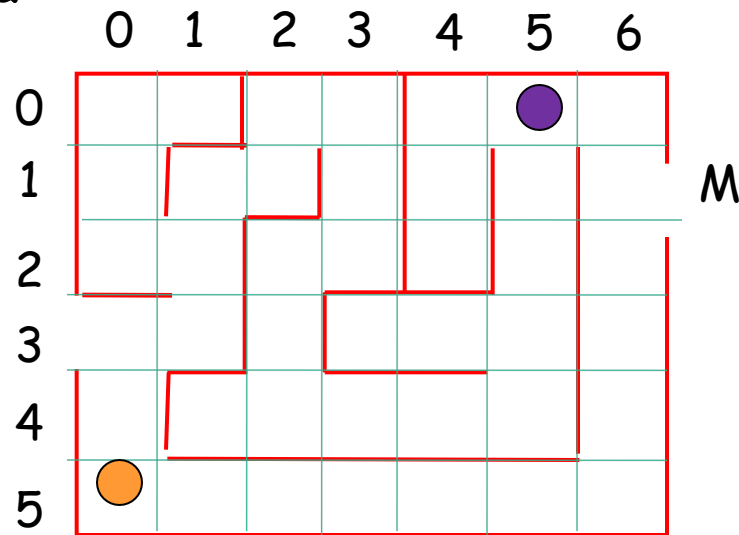
- Definir una heurística
- Aplicar búsqueda avara



¿Cuál está más distante de la meta?  
¿Cómo lo sabe?

# Búsqueda informada

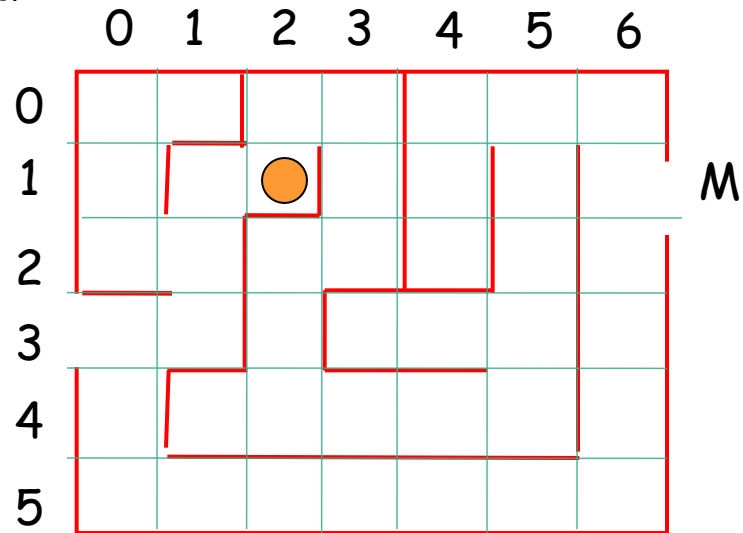
- Definir una heurística
- Aplicar búsqueda avara



$h(n)$ : suma de las distancias horizontales y verticales para alcanzar la salida. Distancia de Manhattan

# Búsqueda informada

- Definir una heurística
- Aplicar búsqueda avara



# Búsqueda informada

## Aplicar búsqueda Avara

**Agente Minero.** Indique el camino que sigue el agente.

No evite devolverse

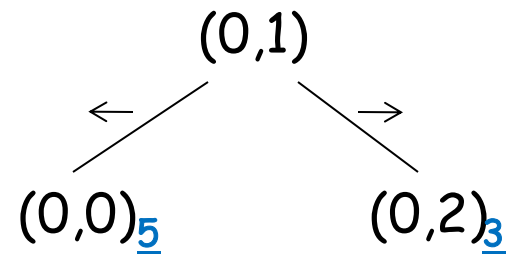


Punto de inicio



Lingote

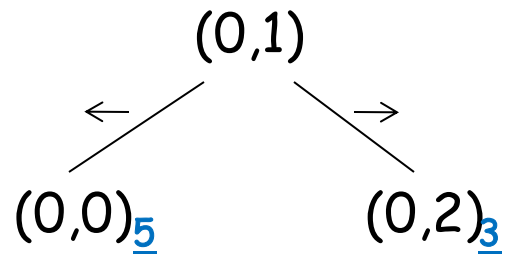
	0	1	2	3
0				
1				
2				



- Utilice Manhattan

# Búsqueda informada

---

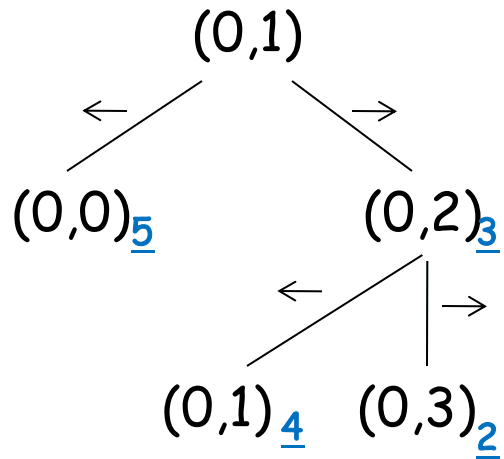


	0	1	2	3
0				
1				
2				



# Búsqueda informada

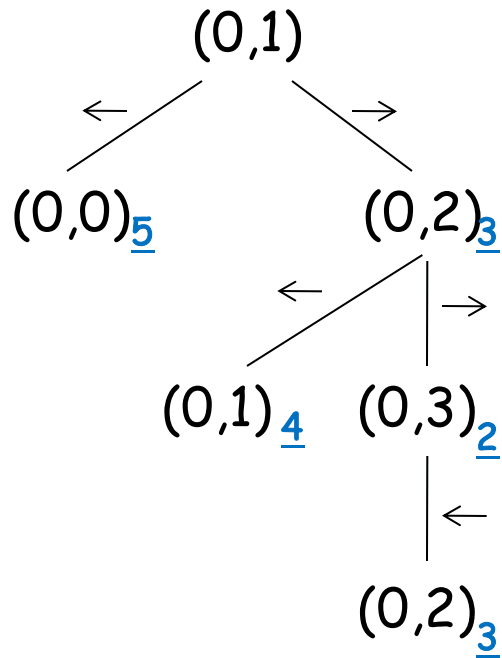
---



	0	1	2	3
0				
1				
2				

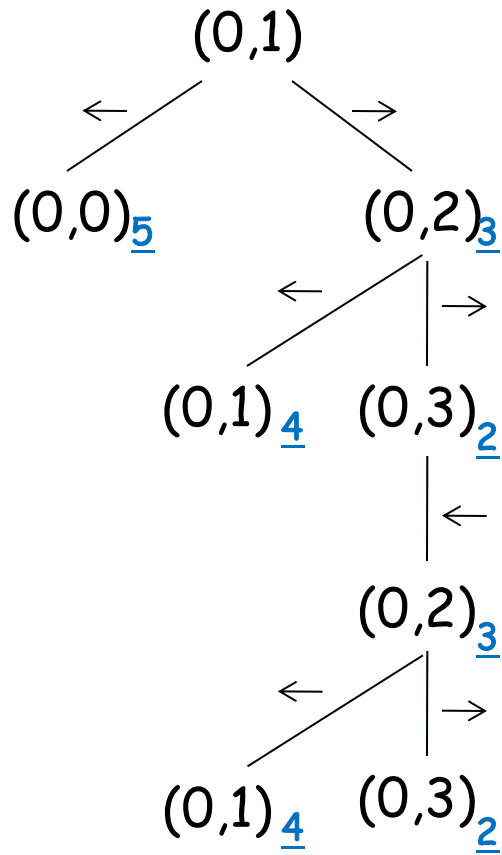
# Búsqueda informada

---



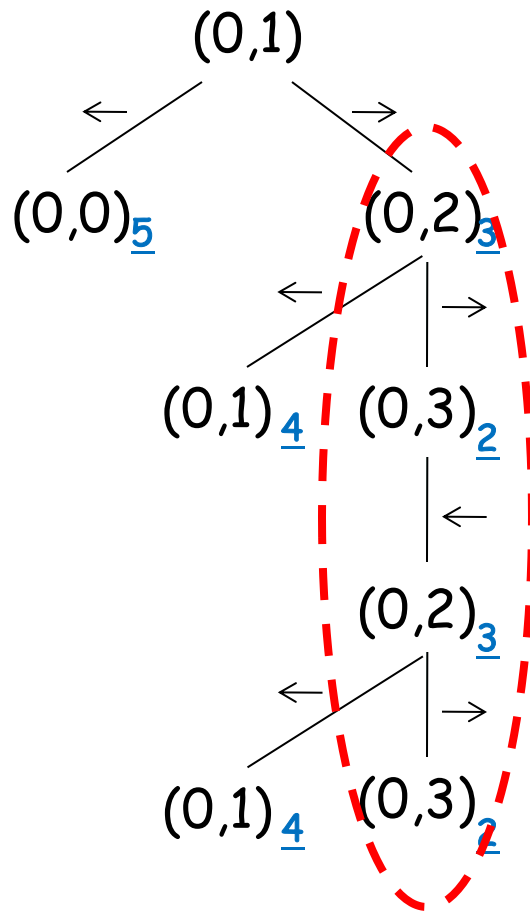
	0	1	2	3
0				
1				
2				

# Búsqueda informada



	0	1	2	3
0				
1				
2				

# Búsqueda informada



	0	1	2	3
0				
1				
2				

Búsqueda avara se queda en un ciclo

# Búsqueda informada









## Aplicar búsqueda Avara

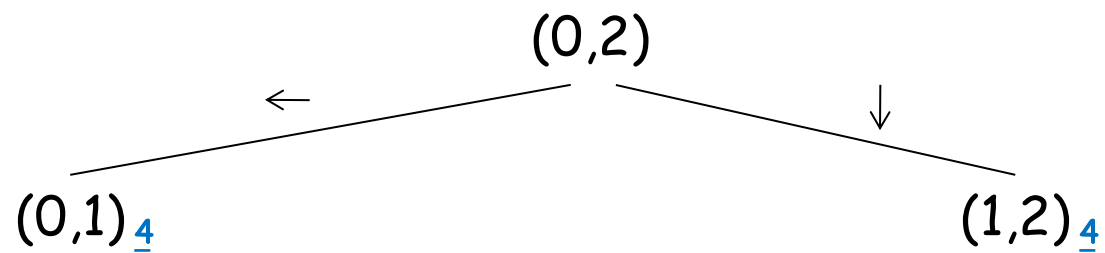
### El ratón inteligente.









- Llegar a una casilla libre cuesta 1
- Llegar a una casilla donde hay gato cuesta 5
- Llegar a una casilla donde hay ratonera cuesta 3

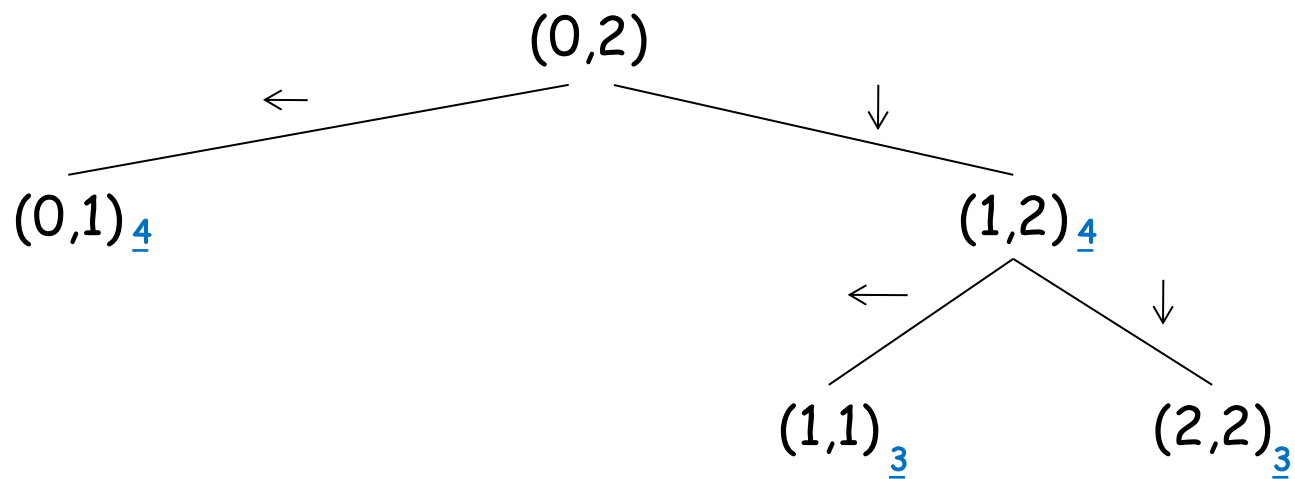
*Evite devolverse*









- Utilice Manhattan

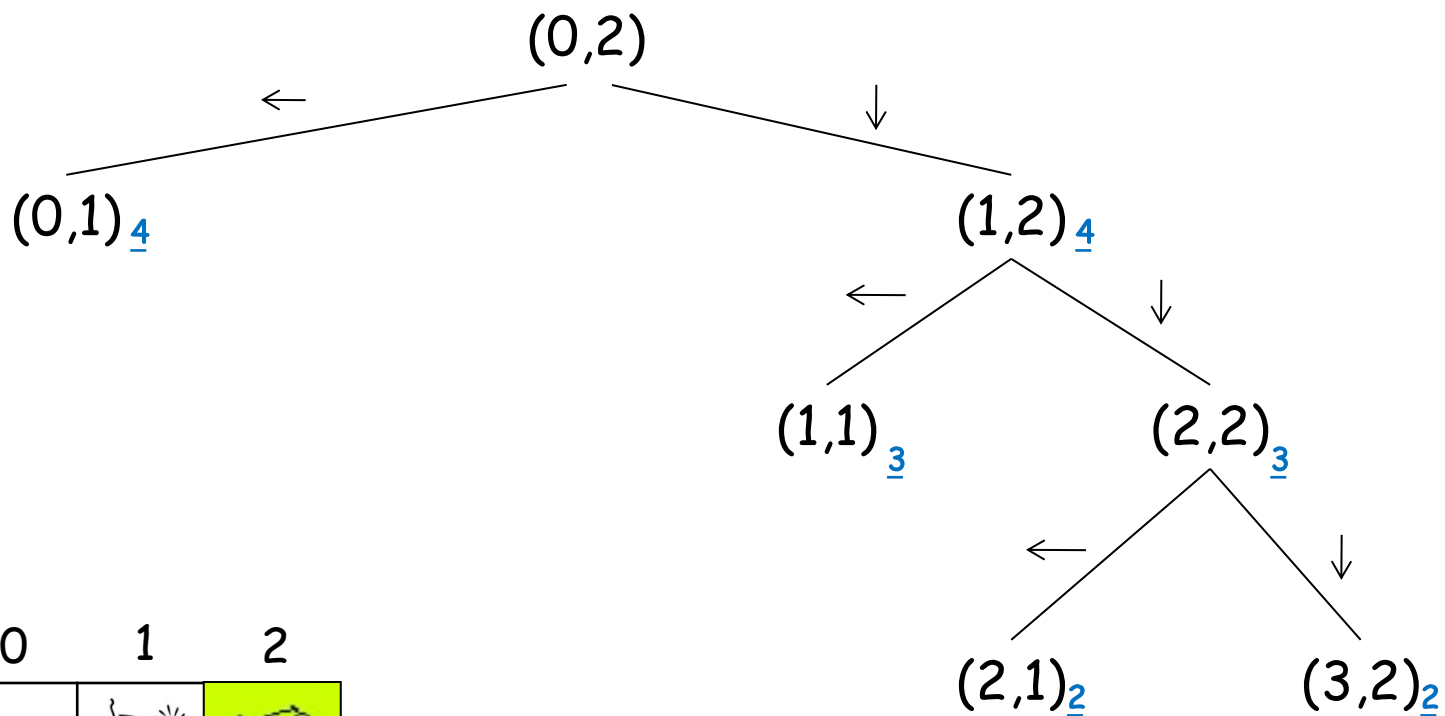
	0	1	2
0			
1			
2			
3			
4			











	0	1	2
0			
1			
2			
3			
4			

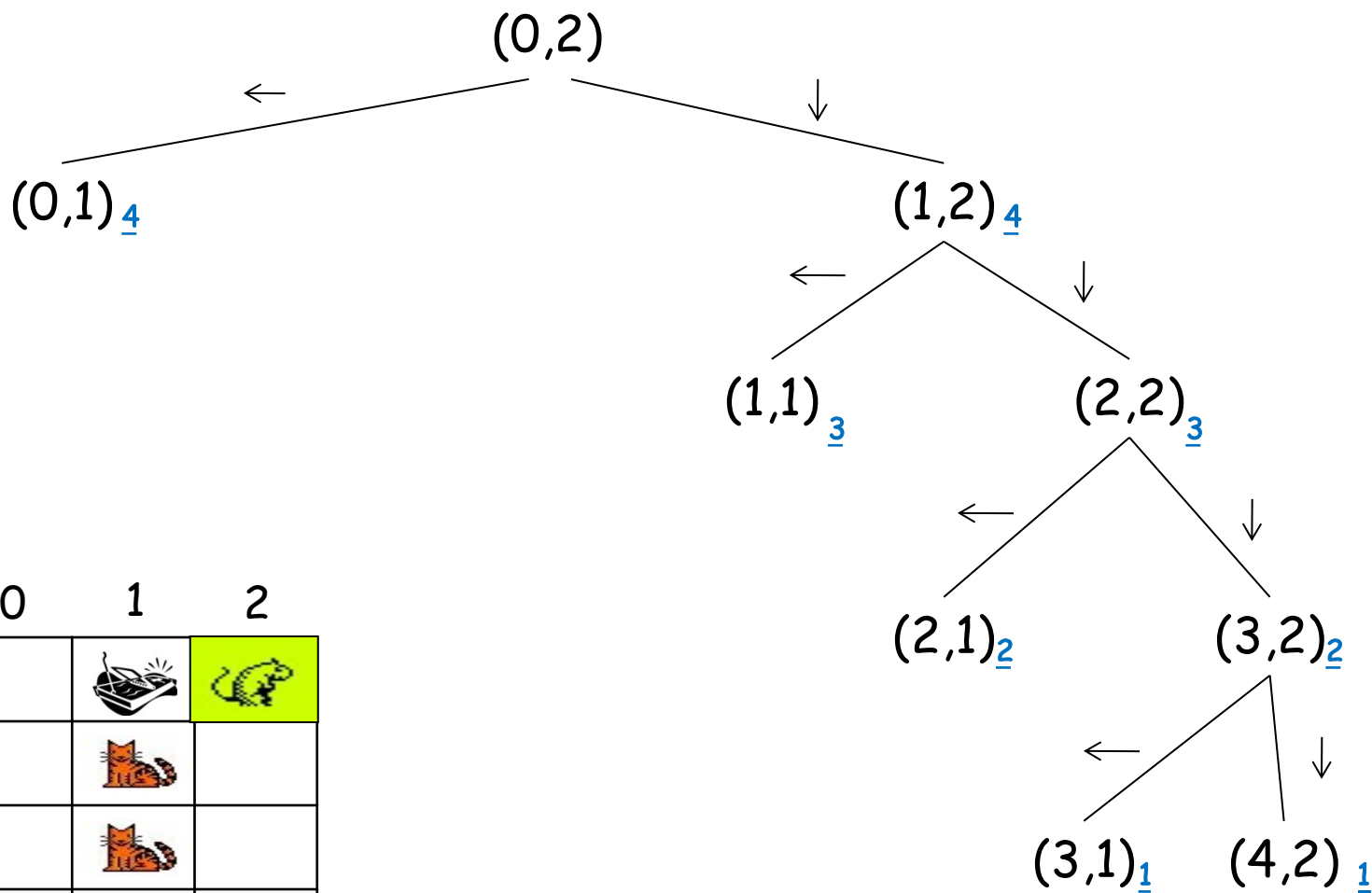










	0	1	2
0			
1			
2			
3			
4			

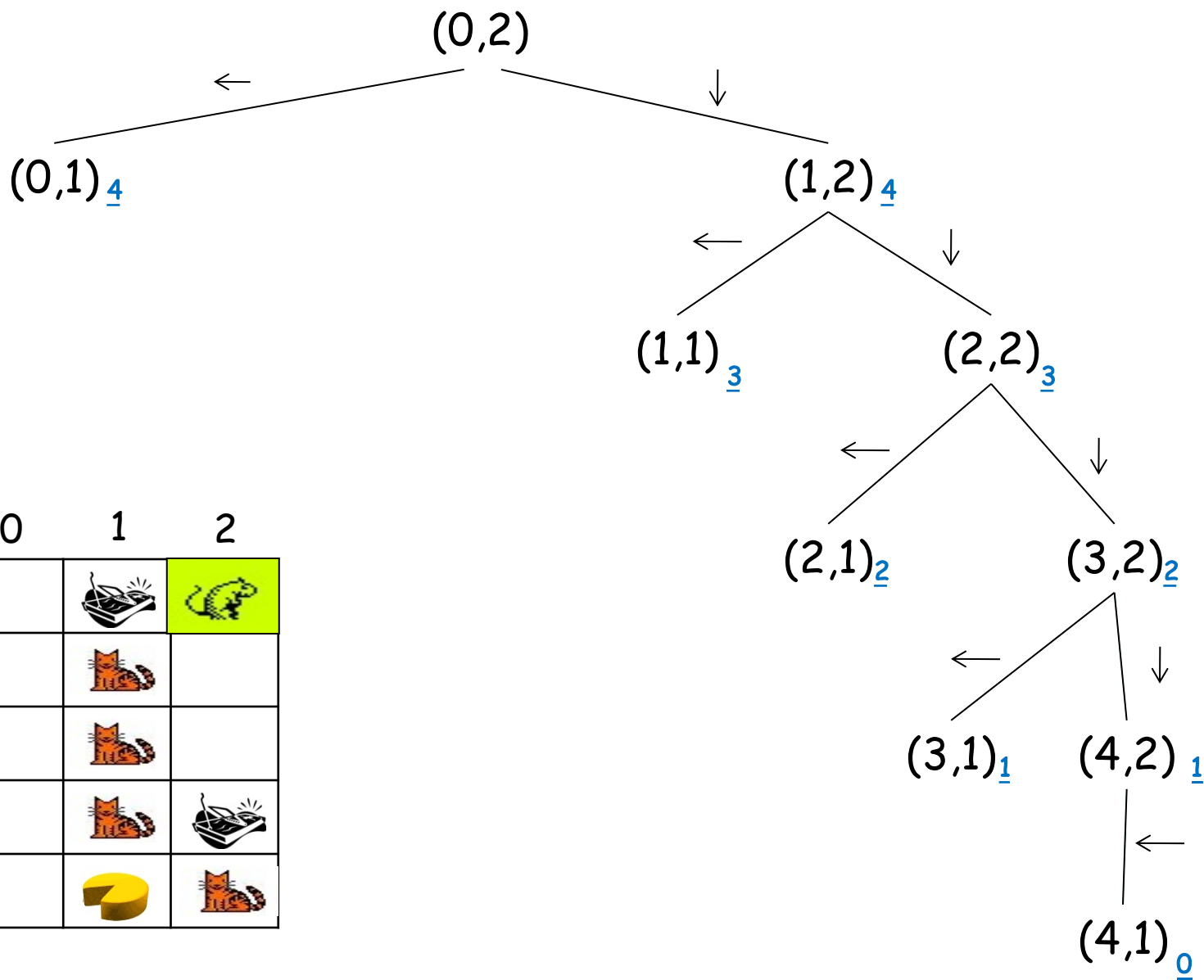










	0	1	2
0			
1			
2			
3			
4			

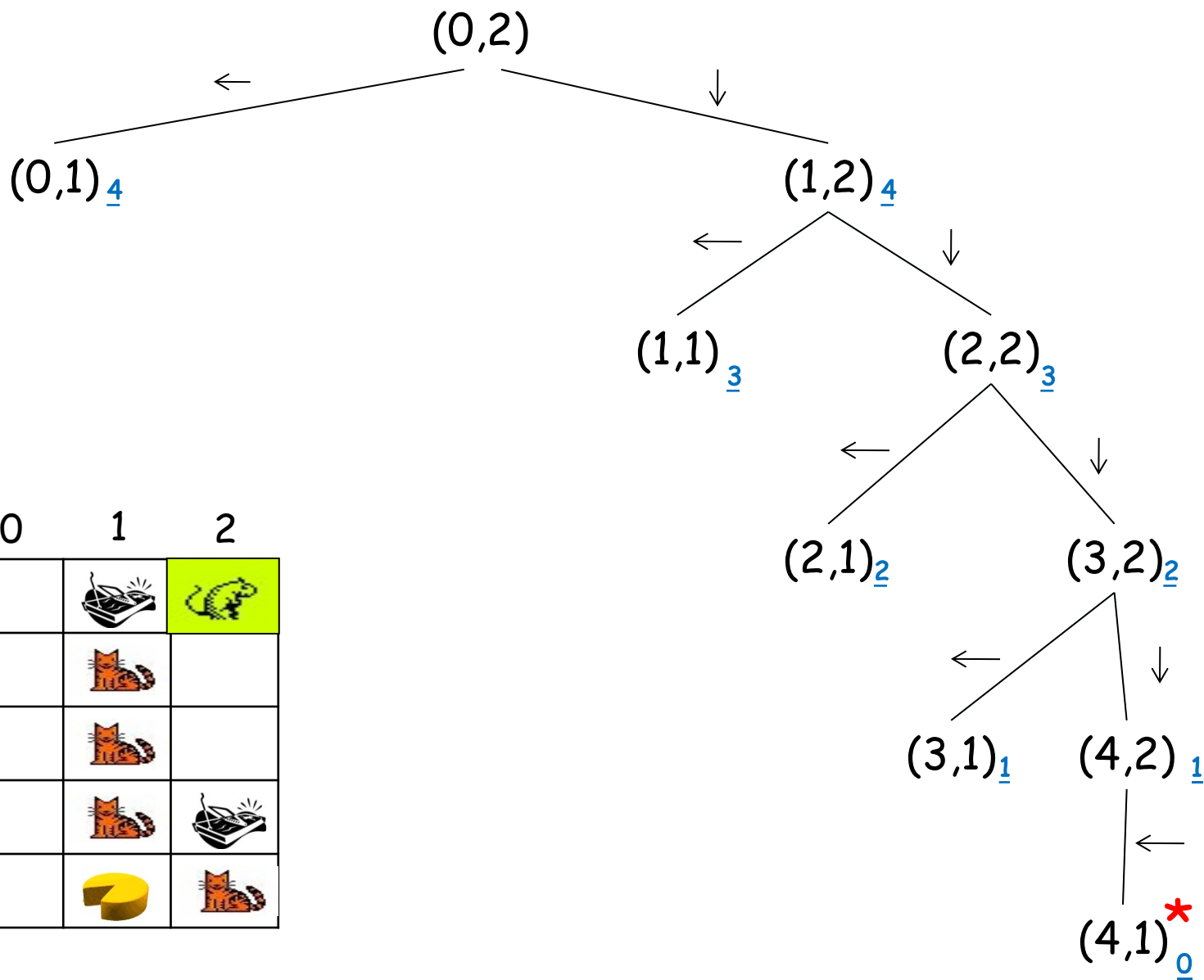












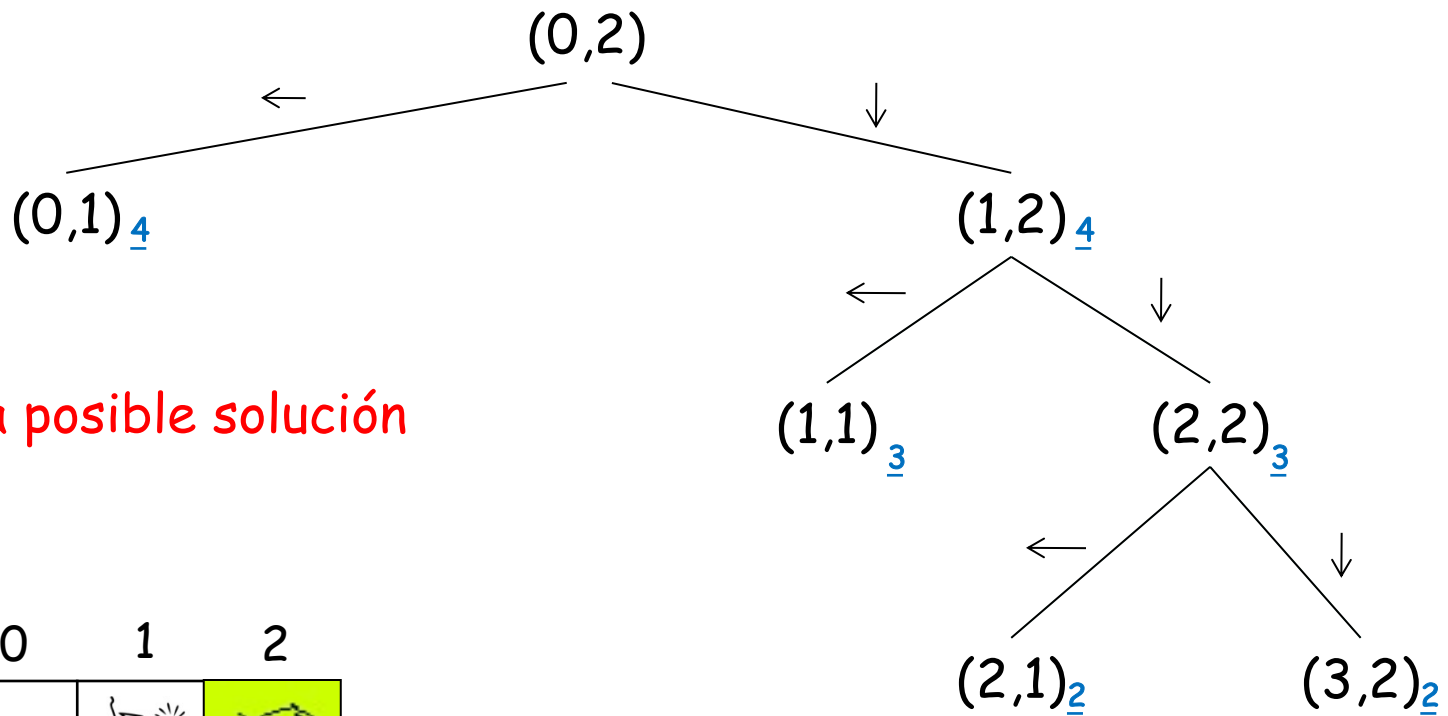
	0	1	2
0			
1			
2			
3			
4			











	0	1	2
0			
1			
2			
3			
4			

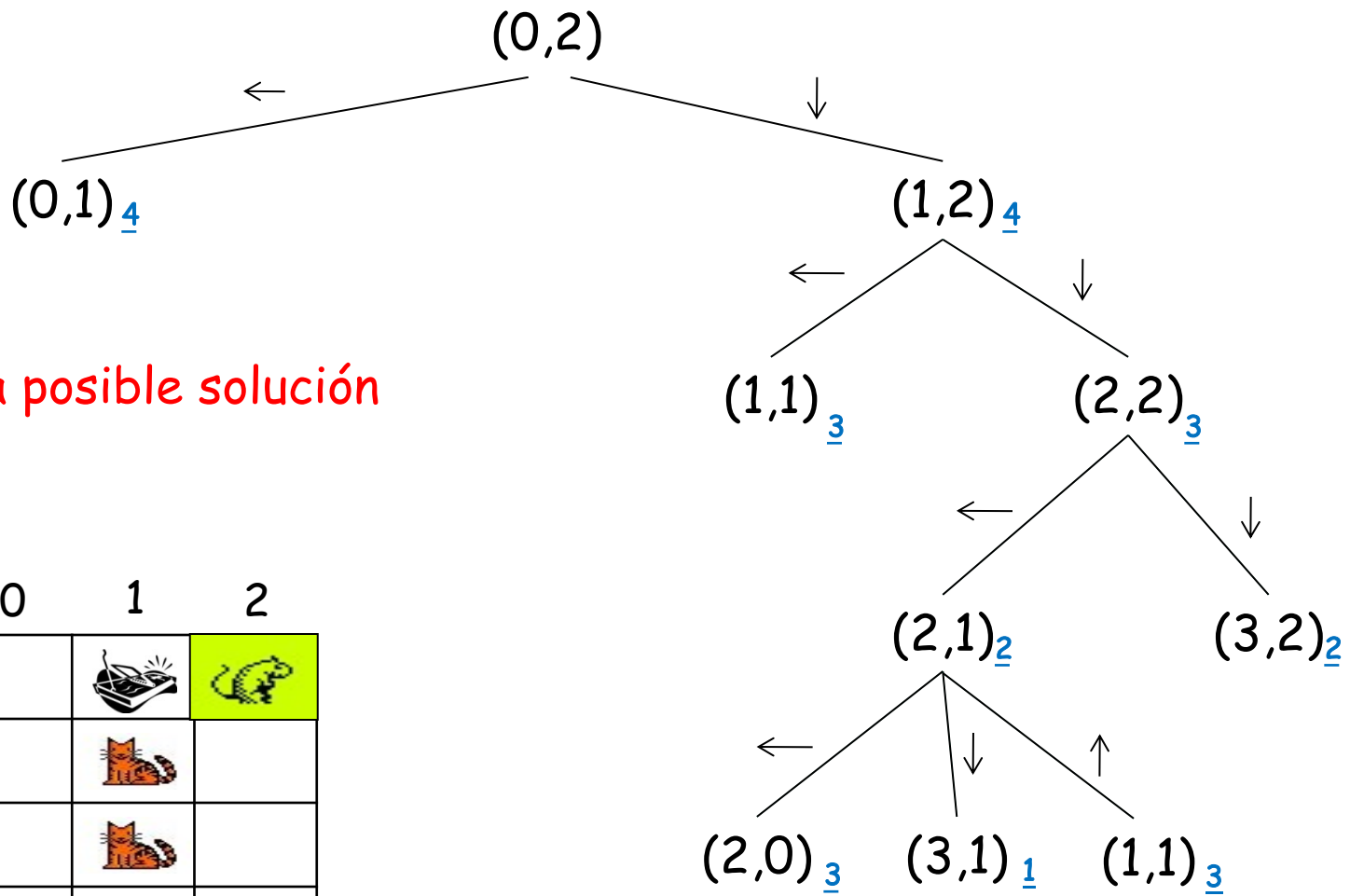


	0	1	2
0			
1			
2			
3			
4			











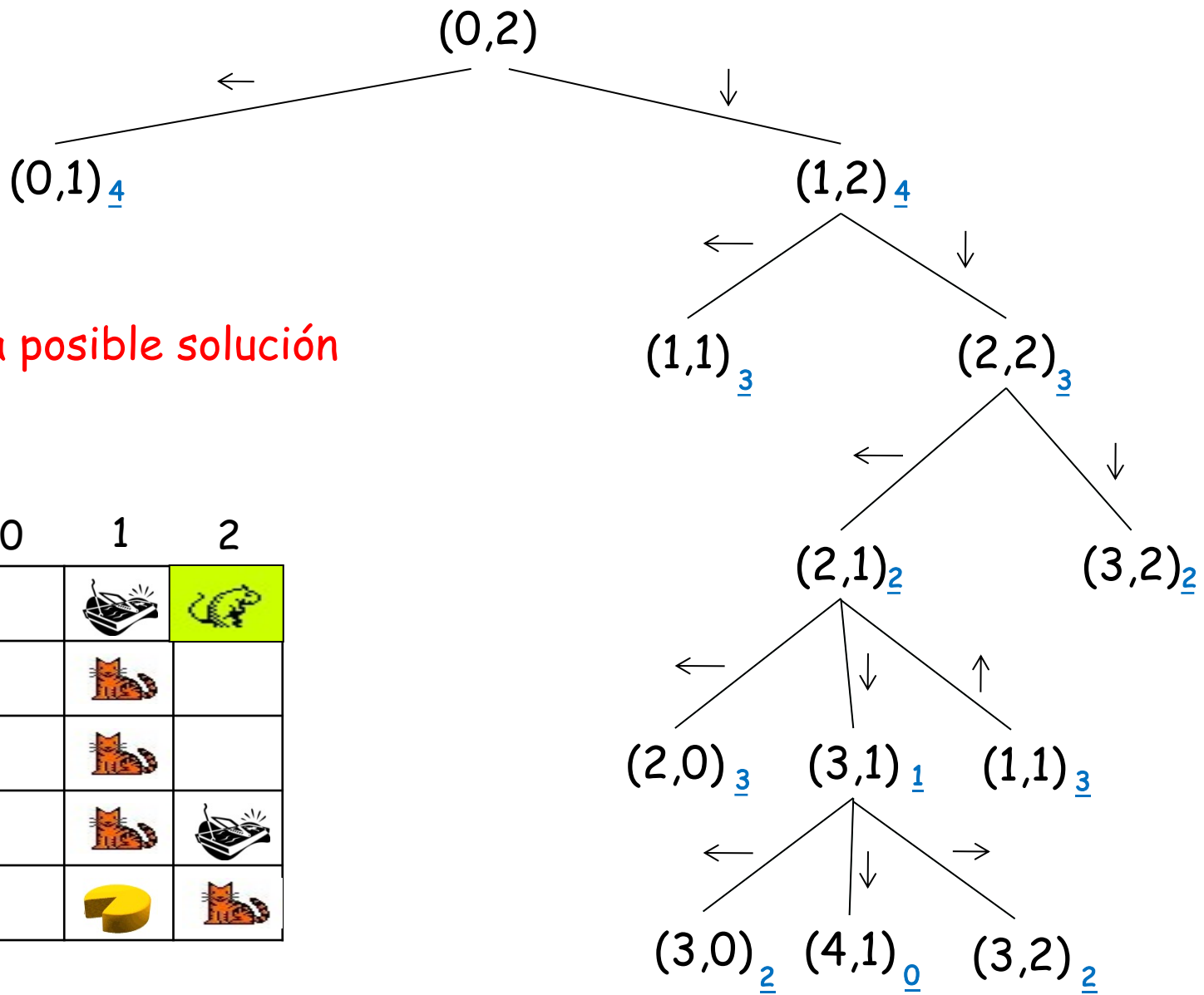
Otra posible solución

	0	1	2
0			
1			
2			
3			
4			

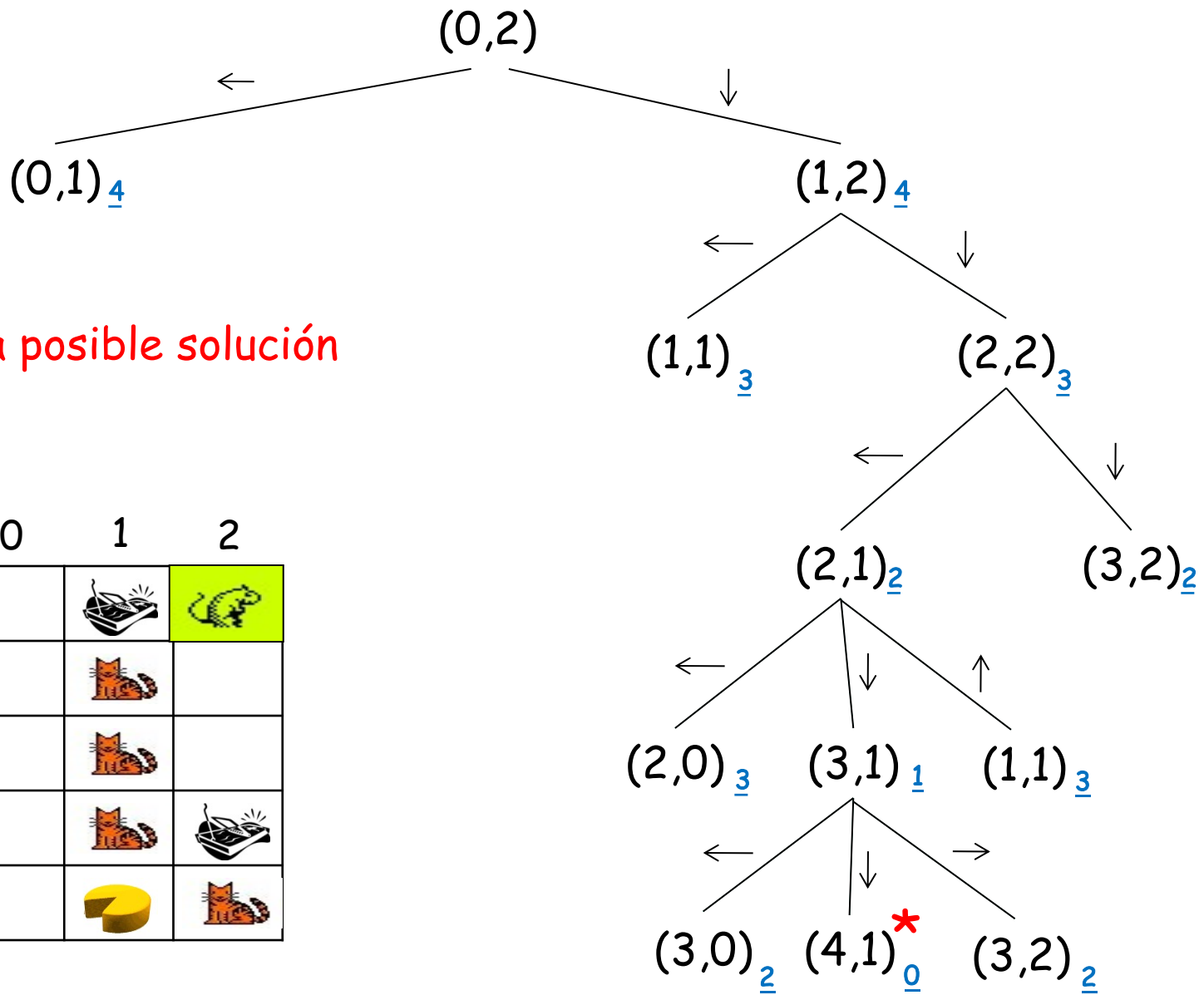


Otra posible solución

	0	1	2
0			
1			
2			
3			
4			



Otra posible solución



Otra posible solución

# Búsqueda informada

---

## Implementación

- Se puede implementar considerando la lista de nodos a expandir como una cola de prioridad, donde la prioridad es el valor de la heurística



# Búsqueda informada

---

- **Completitud:** no, puede caer en ciclos
- **Complejidad temporal:**  $O(b^d)$ , exponencial
- **Complejidad espacial:**  $O(b^d)$ , exponencial
- **Solución óptima:** no, la función heurística no garantiza la solución óptima

# Búsqueda informada

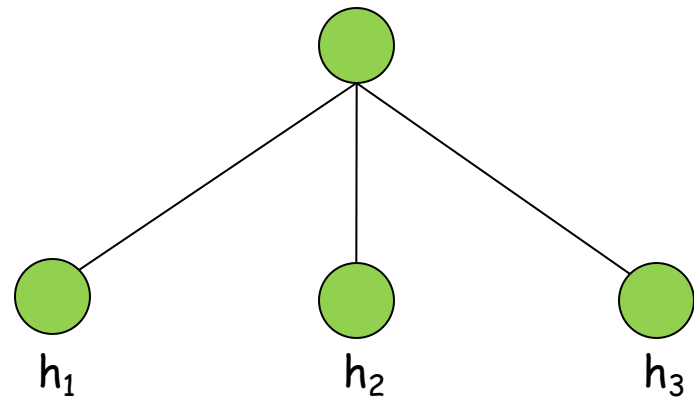
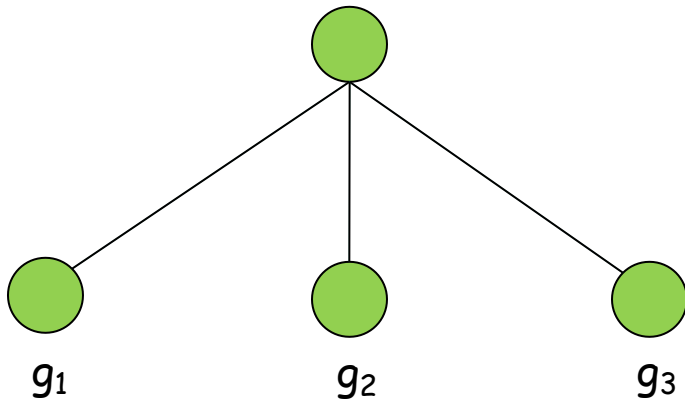
---

- ¿Cuál es la diferencia entre costo uniforme y avara?

# Búsqueda informada

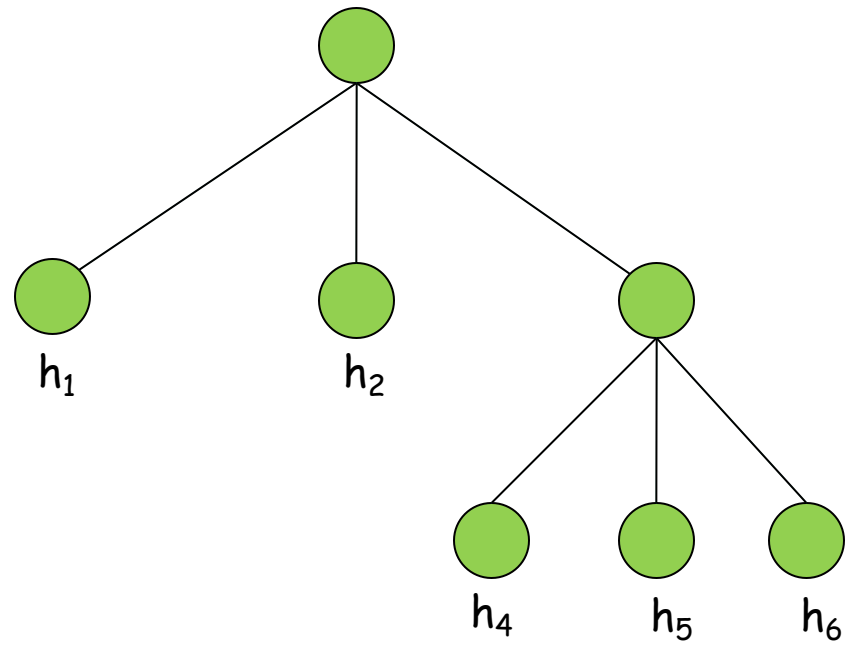
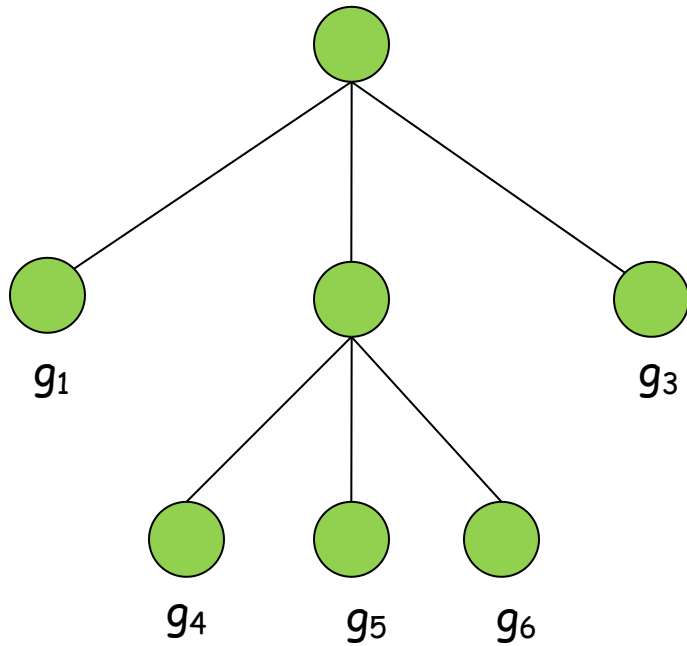
---

- ¿Cuál es la diferencia entre costo uniforme y avara?

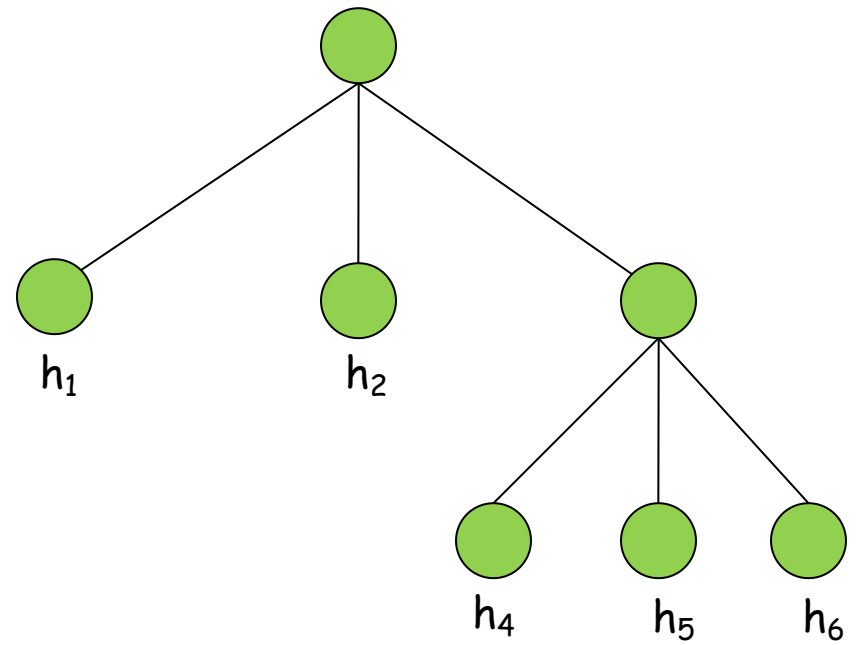
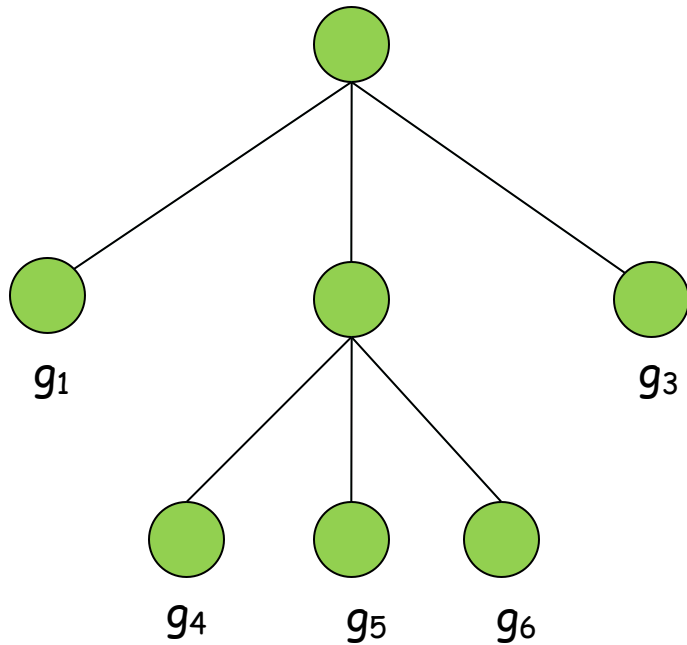













# Búsqueda informada

- ¿Cuál es la diferencia entre costo uniforme y avara?

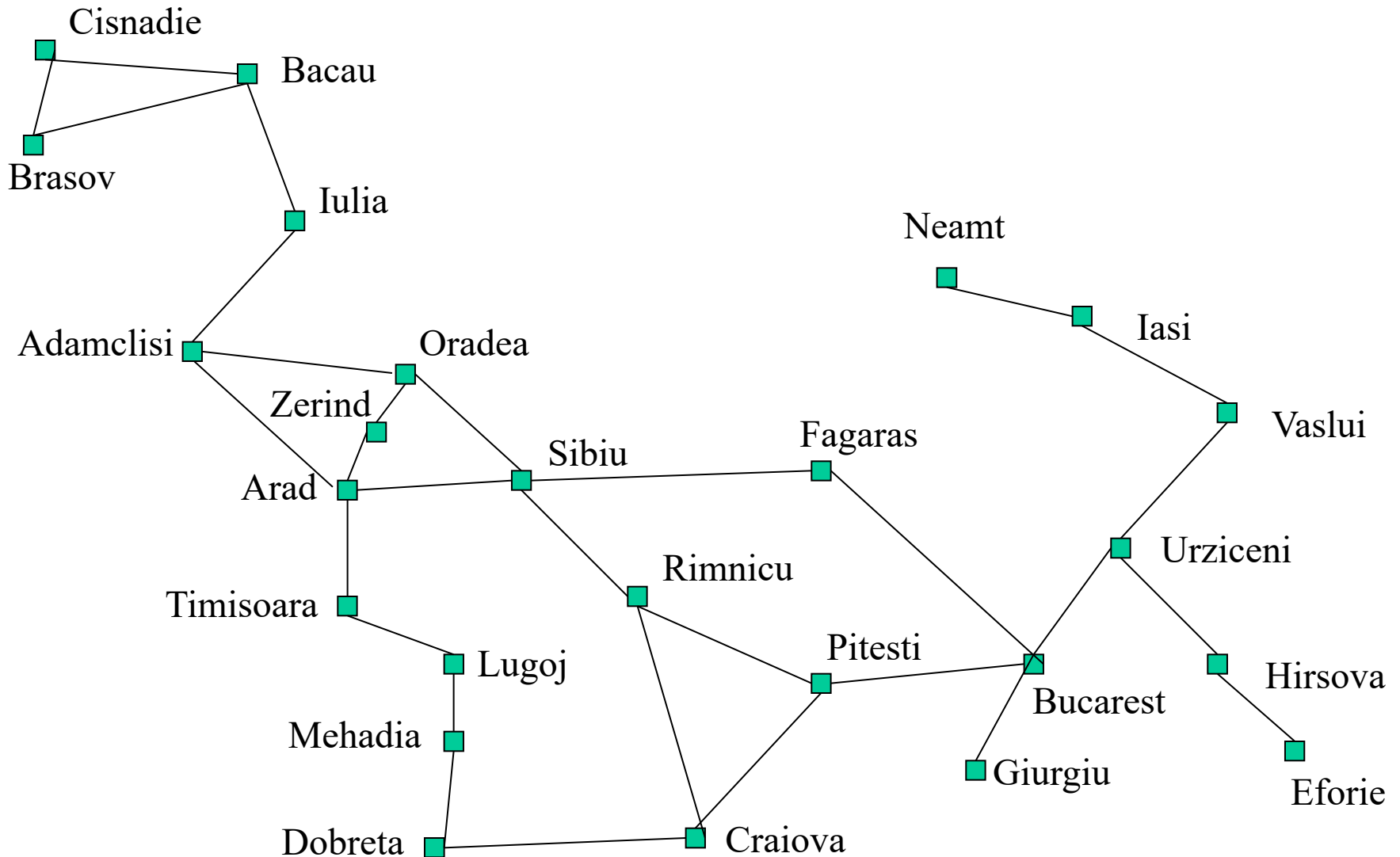


# Búsqueda informada

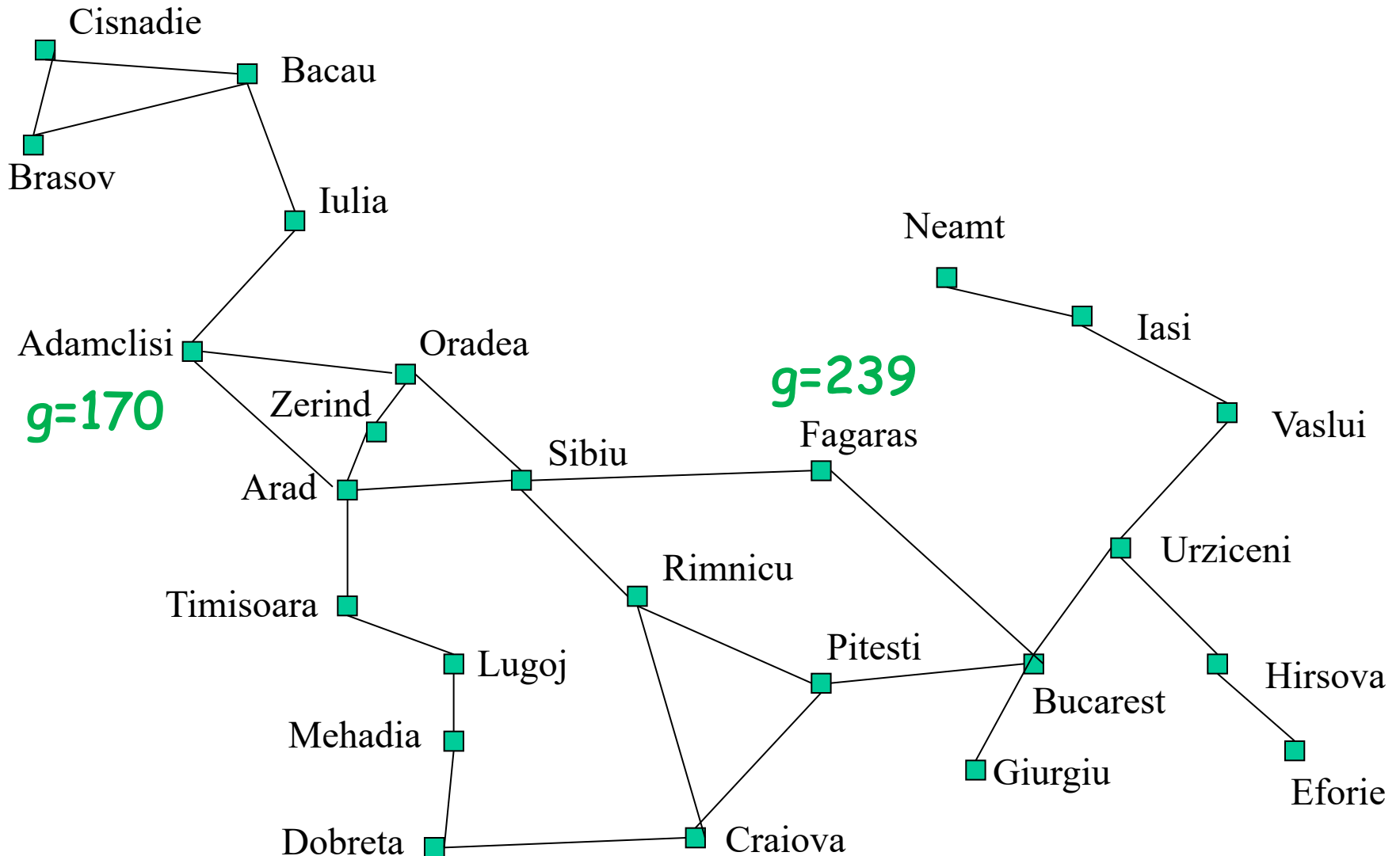


	0	1	2	3
0				
1				
2				
3				
4				

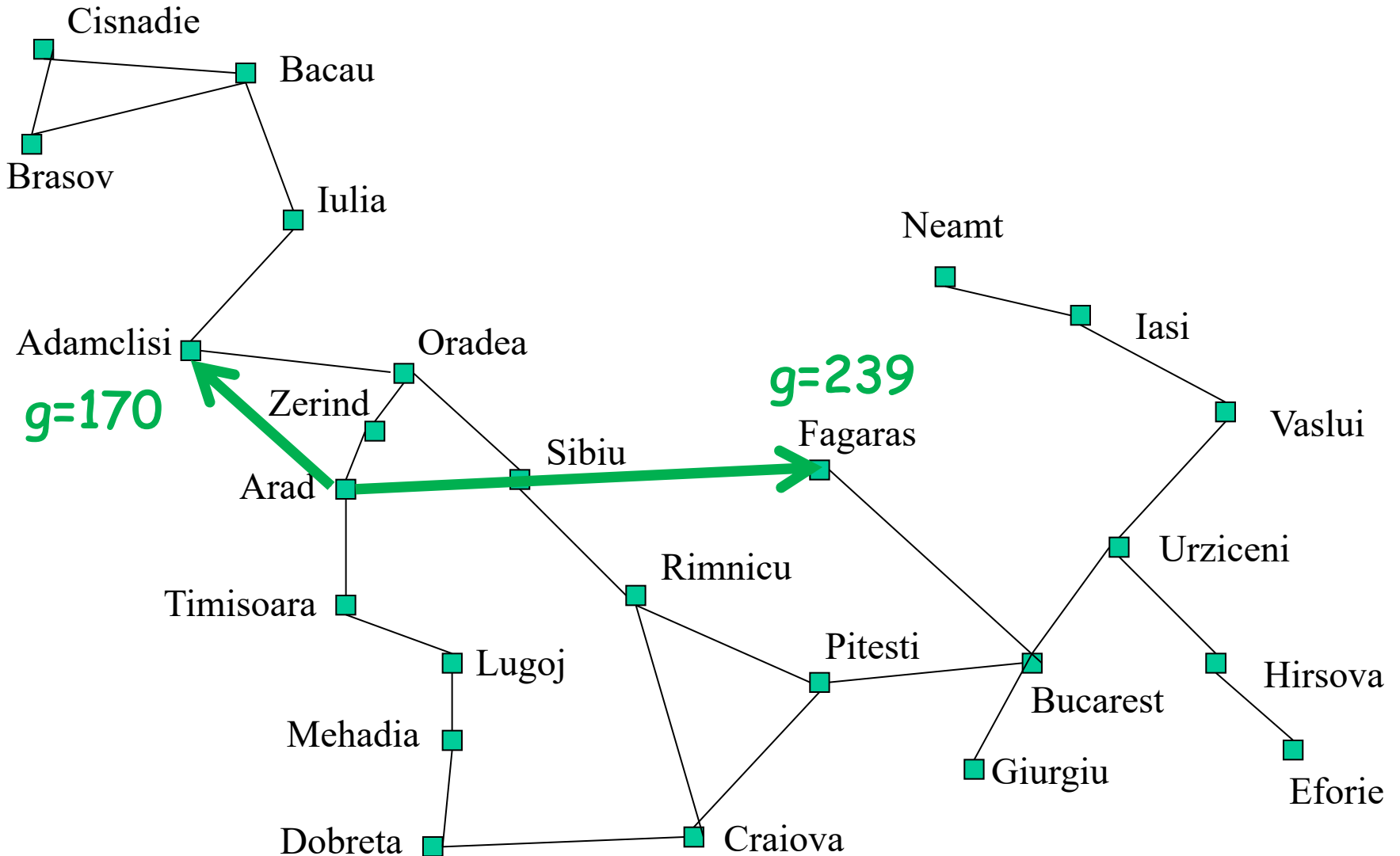
# Búsqueda informada



# Búsqueda informada

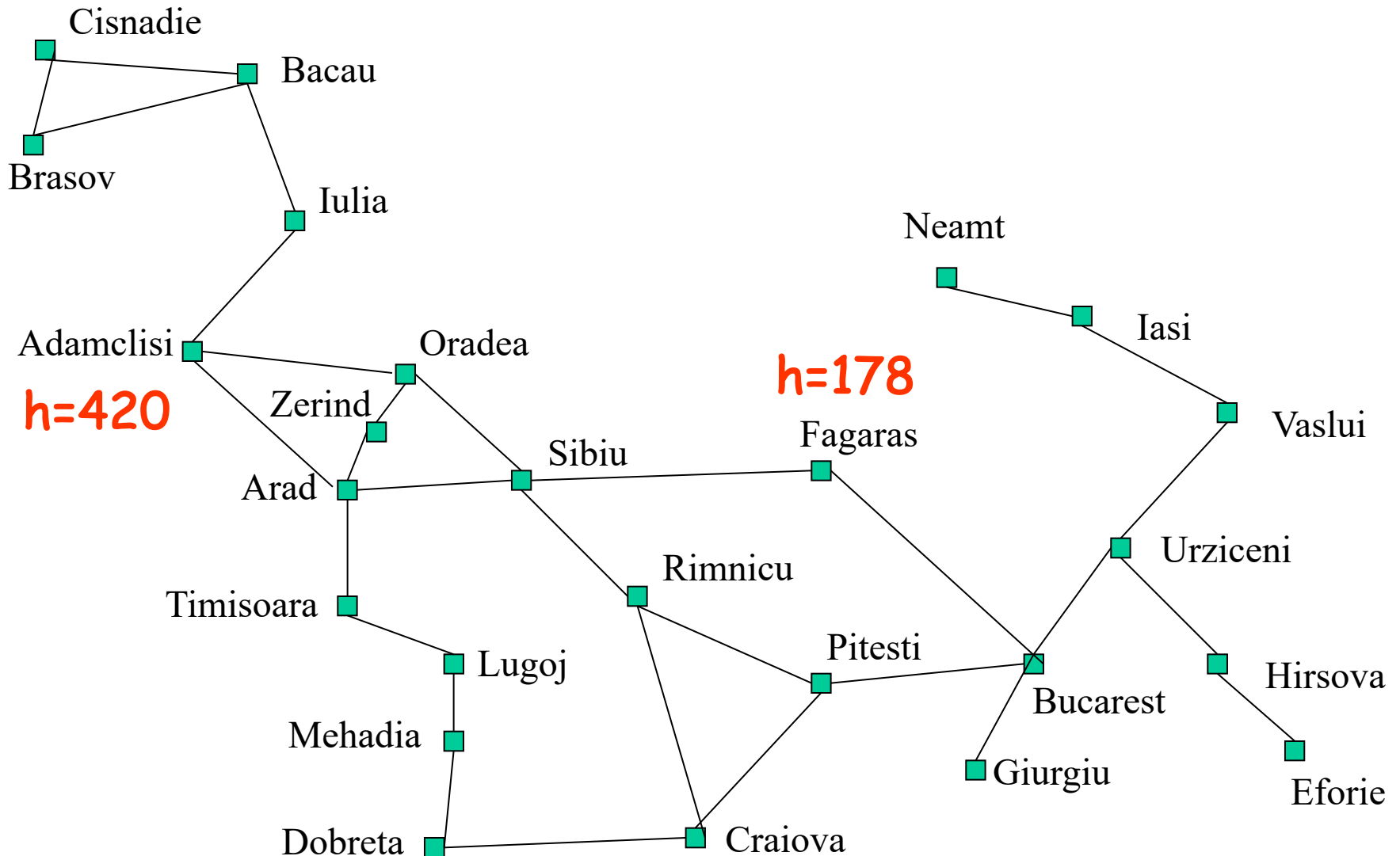


# Búsqueda informada

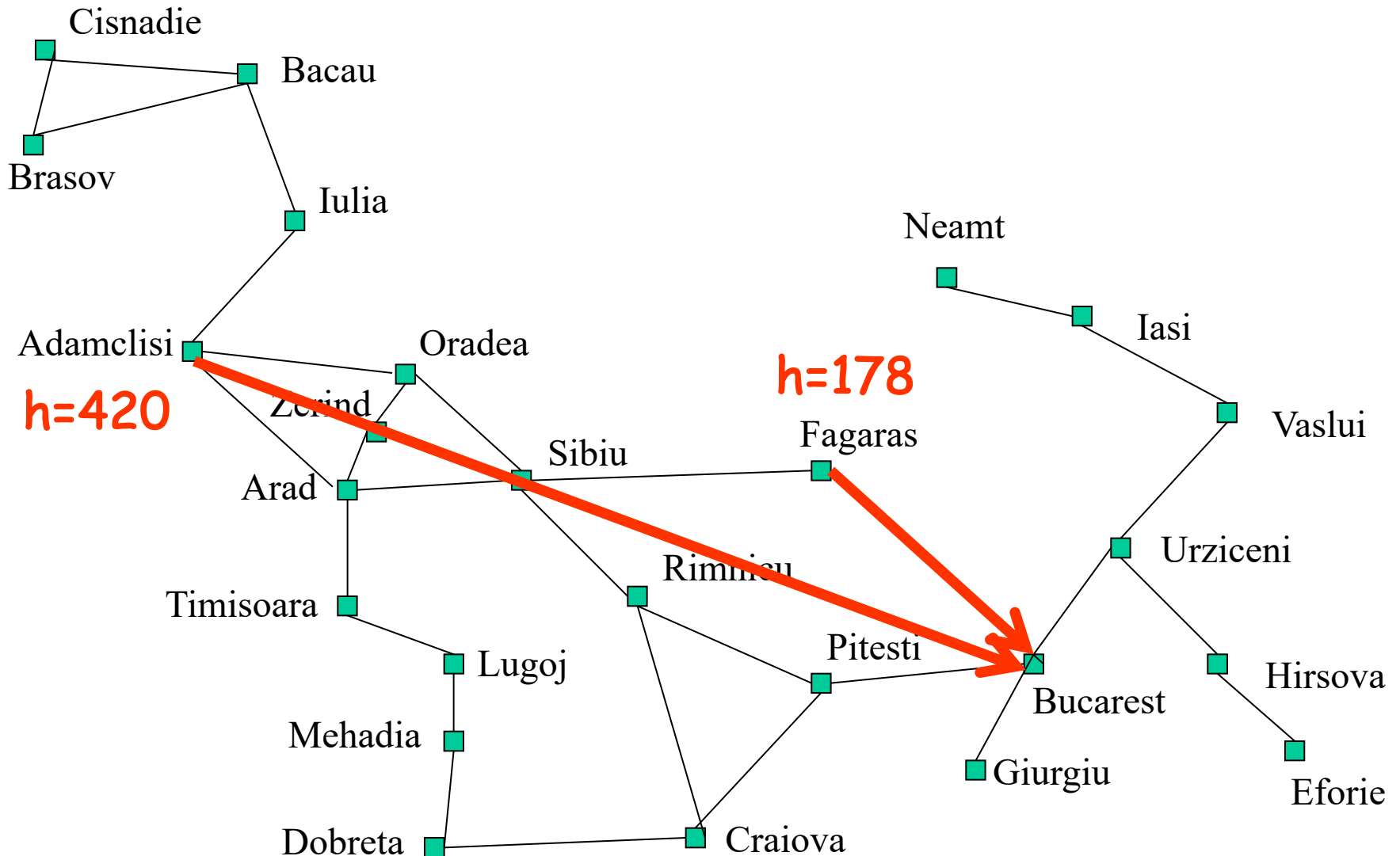




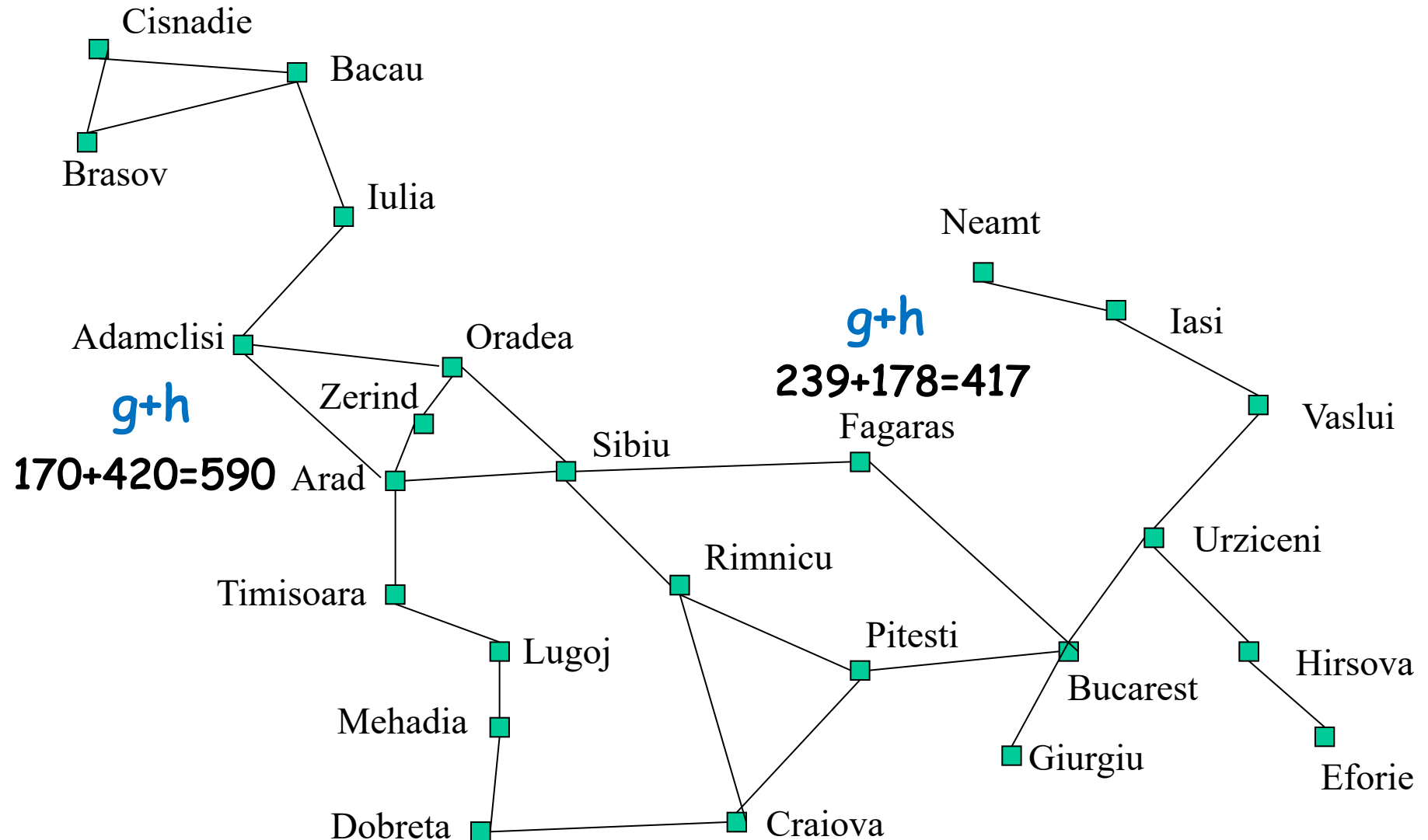
# Búsqueda informada



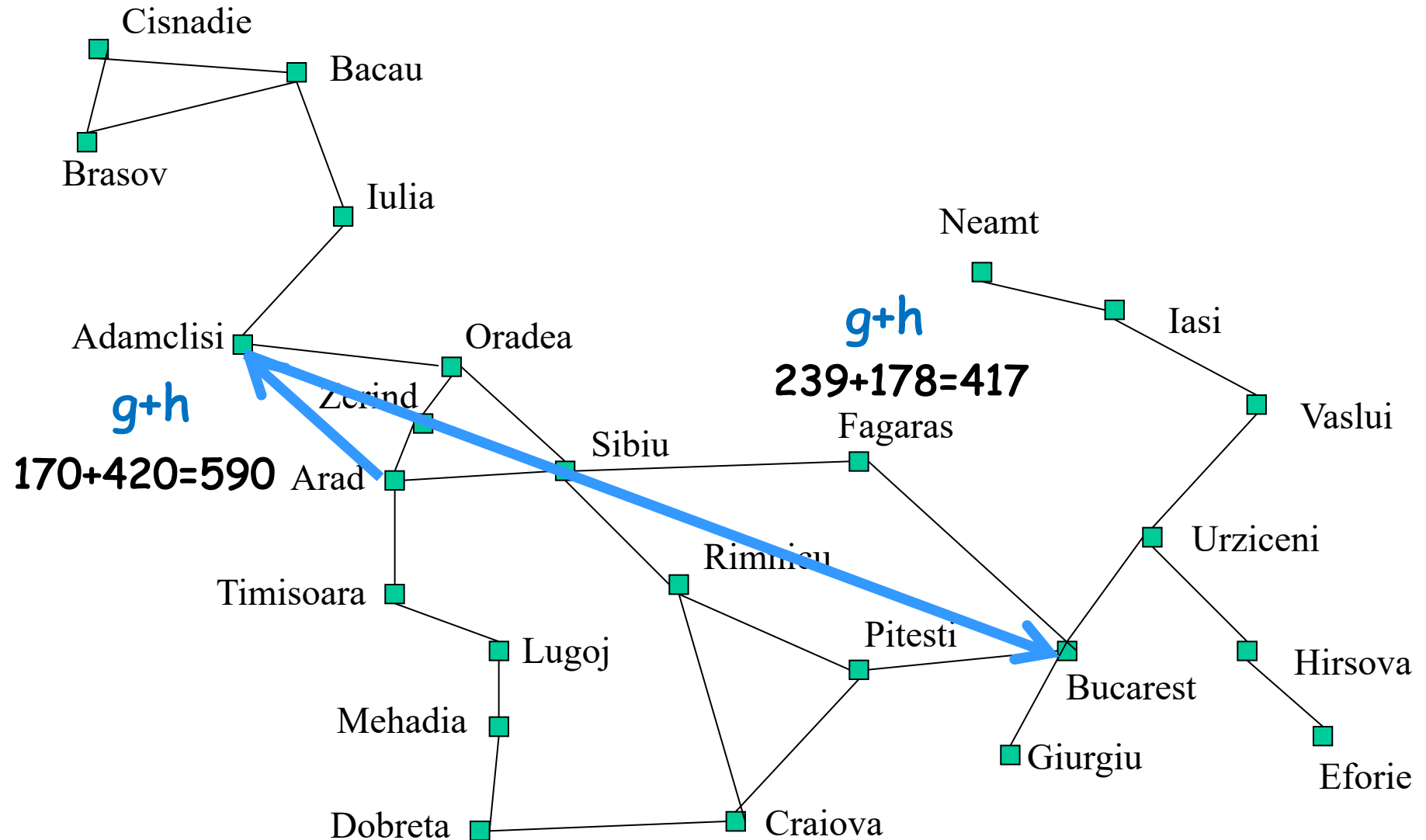
# Búsqueda informada



# Búsqueda informada



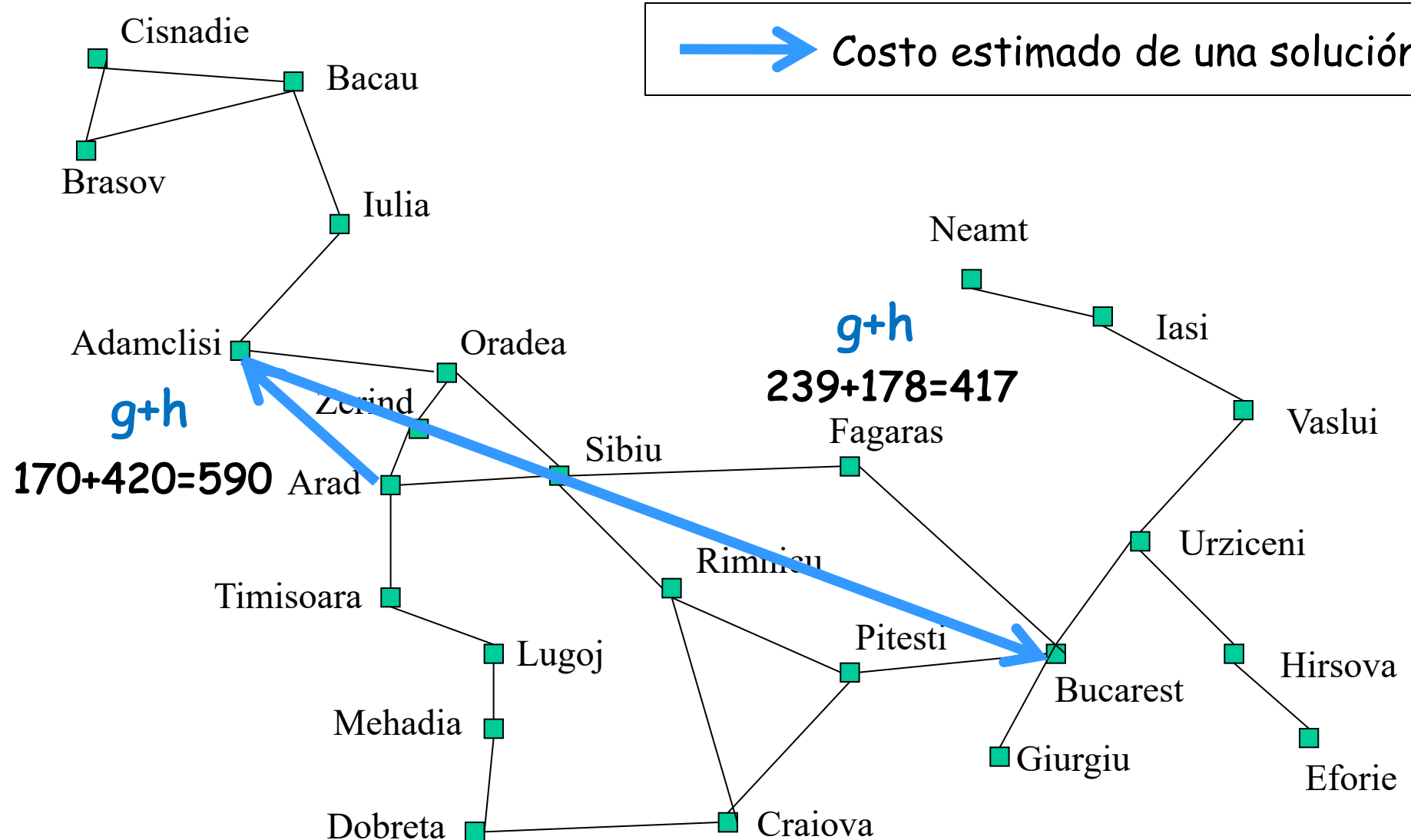
# Búsqueda informada



# Búsqueda informada



Costo estimado de una solución



# Búsqueda informada

---

$g = 50.000, h = 100.000$



$g = 40.000, h = 120.000$



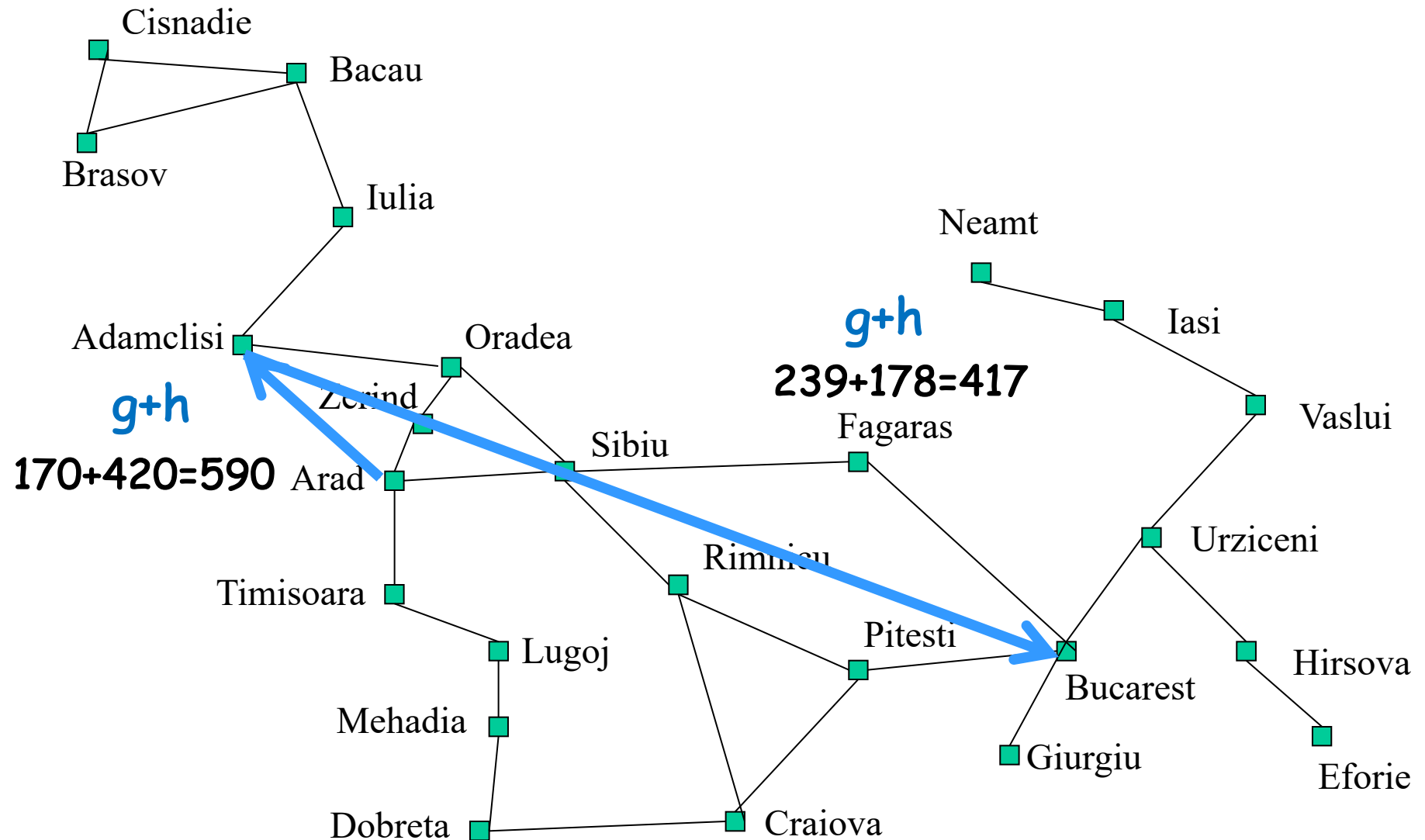
$g = 80.000, h = 100.000$



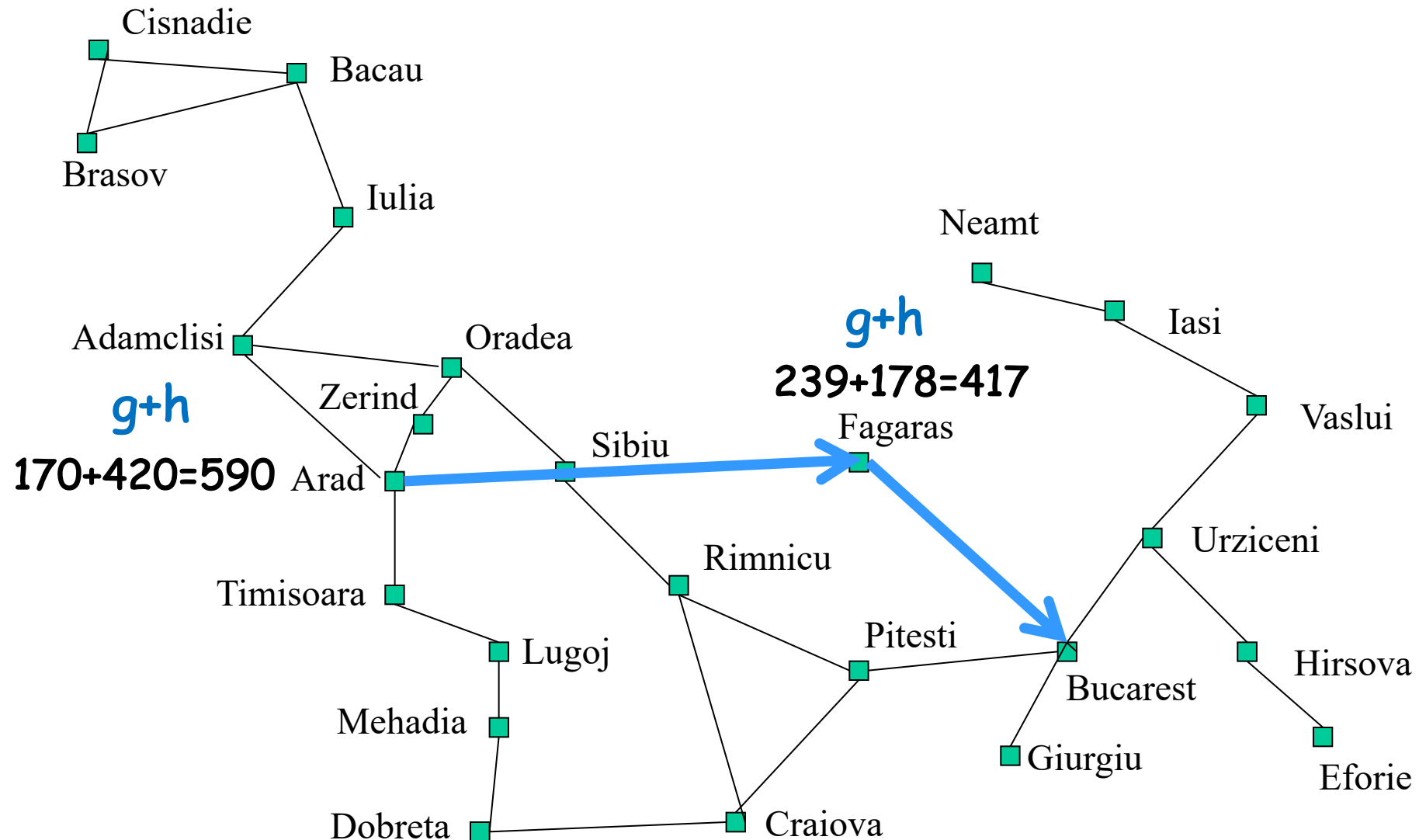
$g = 480.000, h = 20.000$



# Búsqueda informada



# Búsqueda informada





# Búsqueda informada

---

## Algoritmo A\*

- Expandir el nodo con menor  $f(n)$ , esto es, aquel con el menor costo estimado para la solución

$$f(n) = g(n) + h(n)$$

# Búsqueda informada

## Aplicar A\*

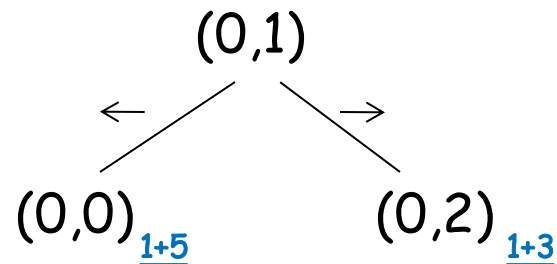
**Agente Minero.** Indique el camino que sigue el agente.

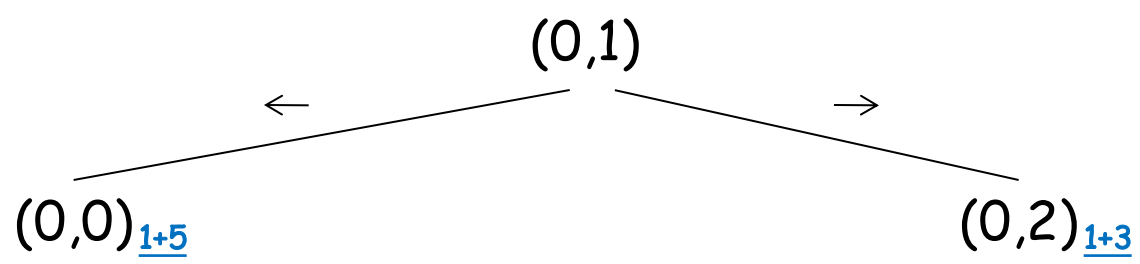
*No evite devolverse*

 Punto de inicio

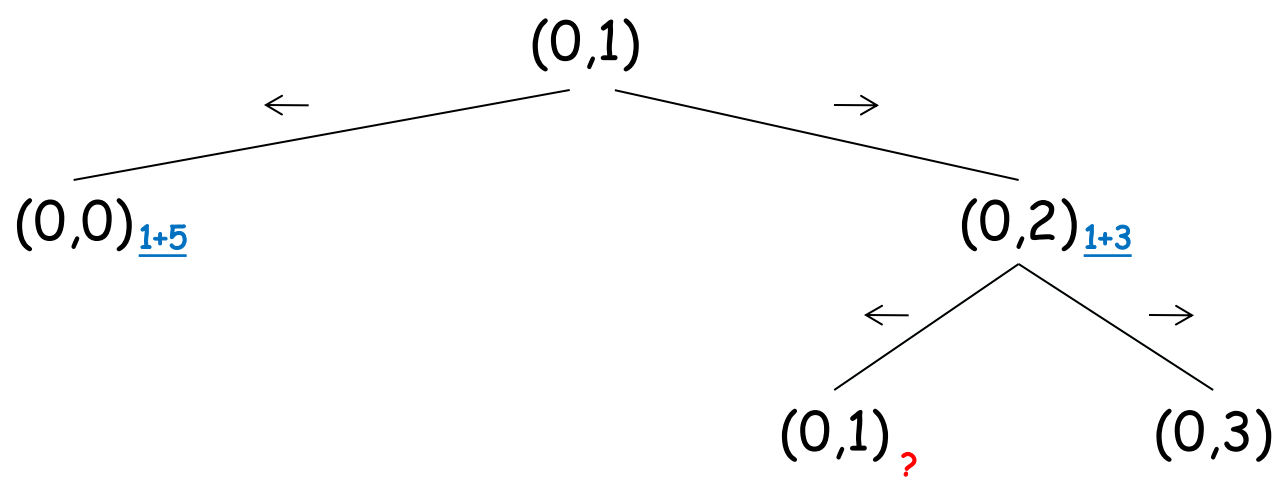
 Lingote

	0	1	2	3
0				
1				
2				

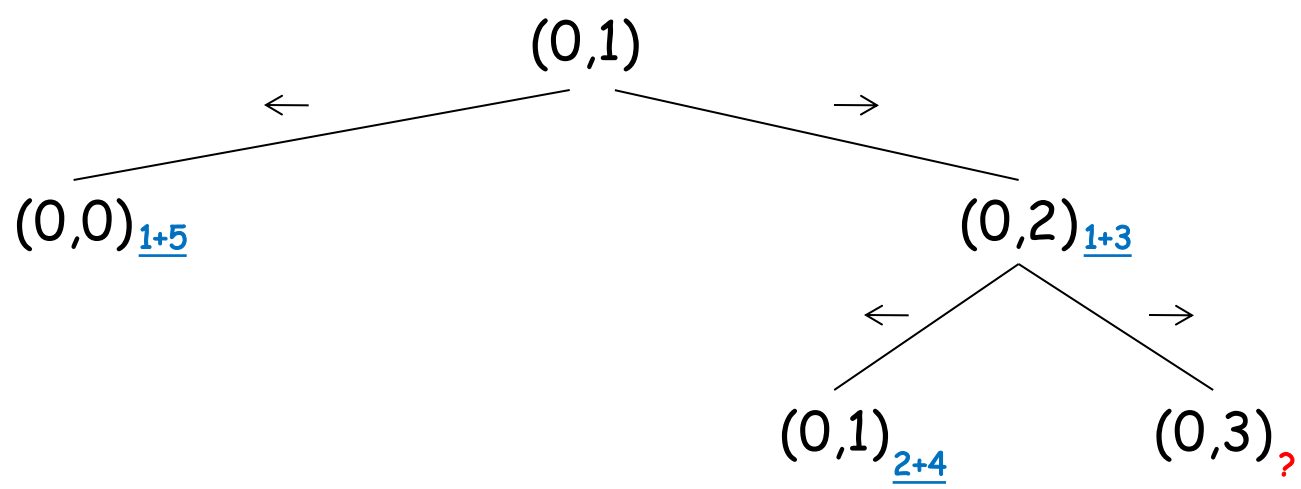




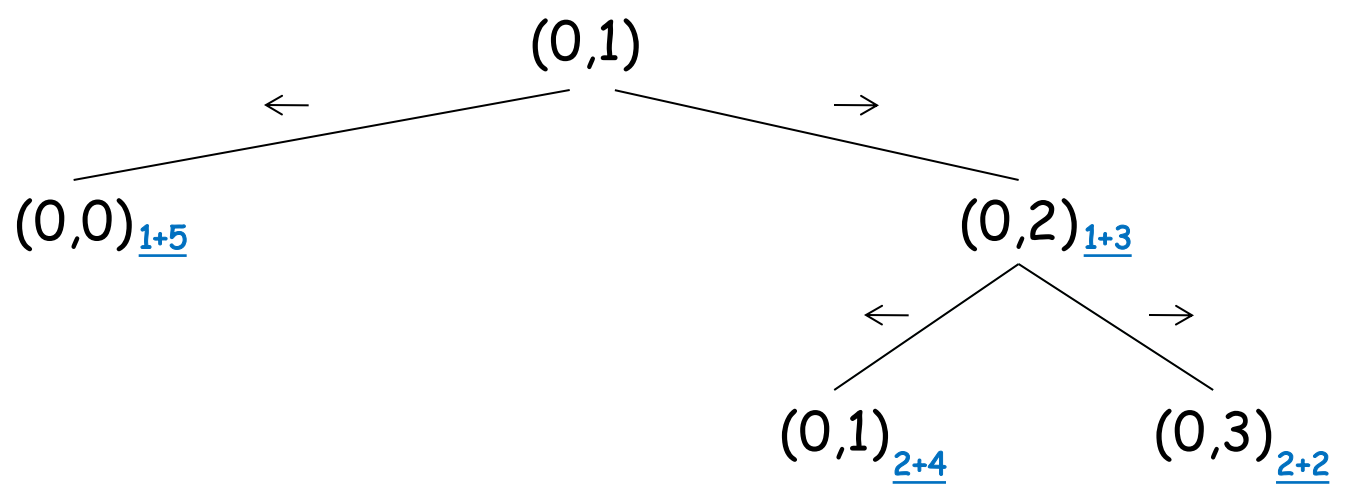
	0	1	2	3
0				
1				
2				



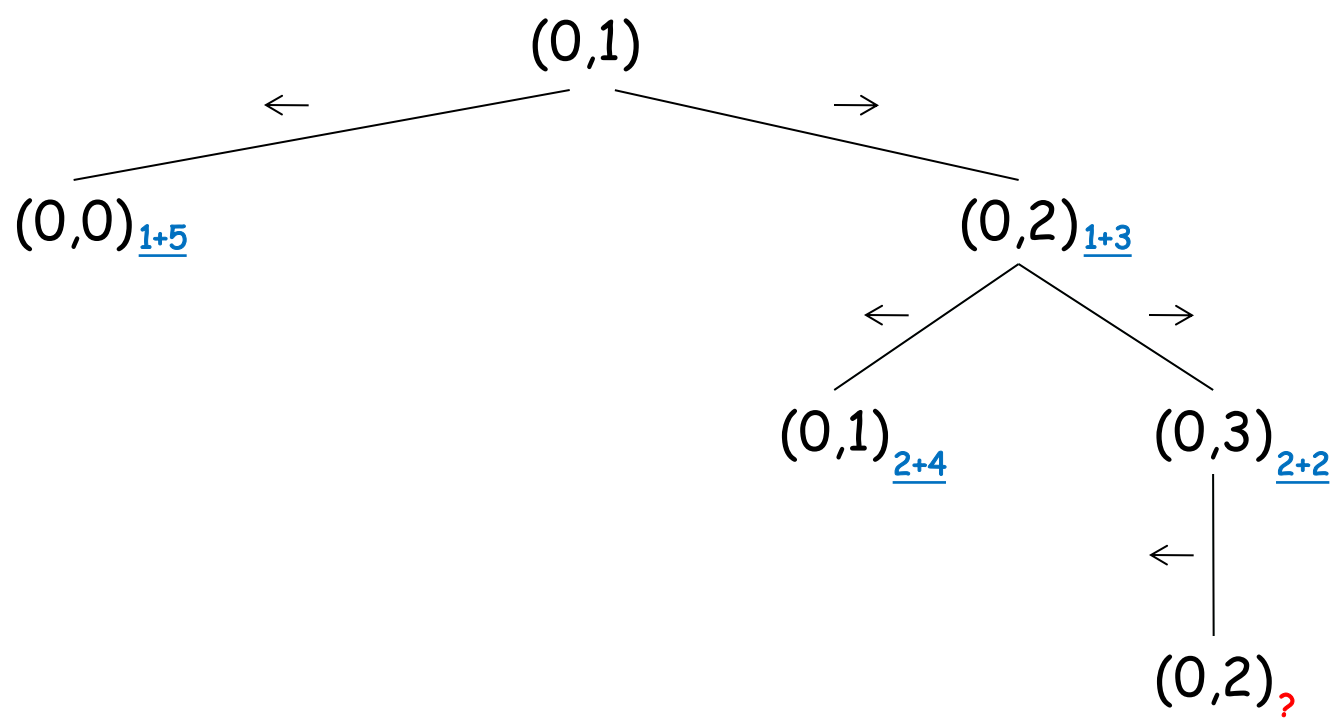
	0	1	2	3
0				
1				
2				



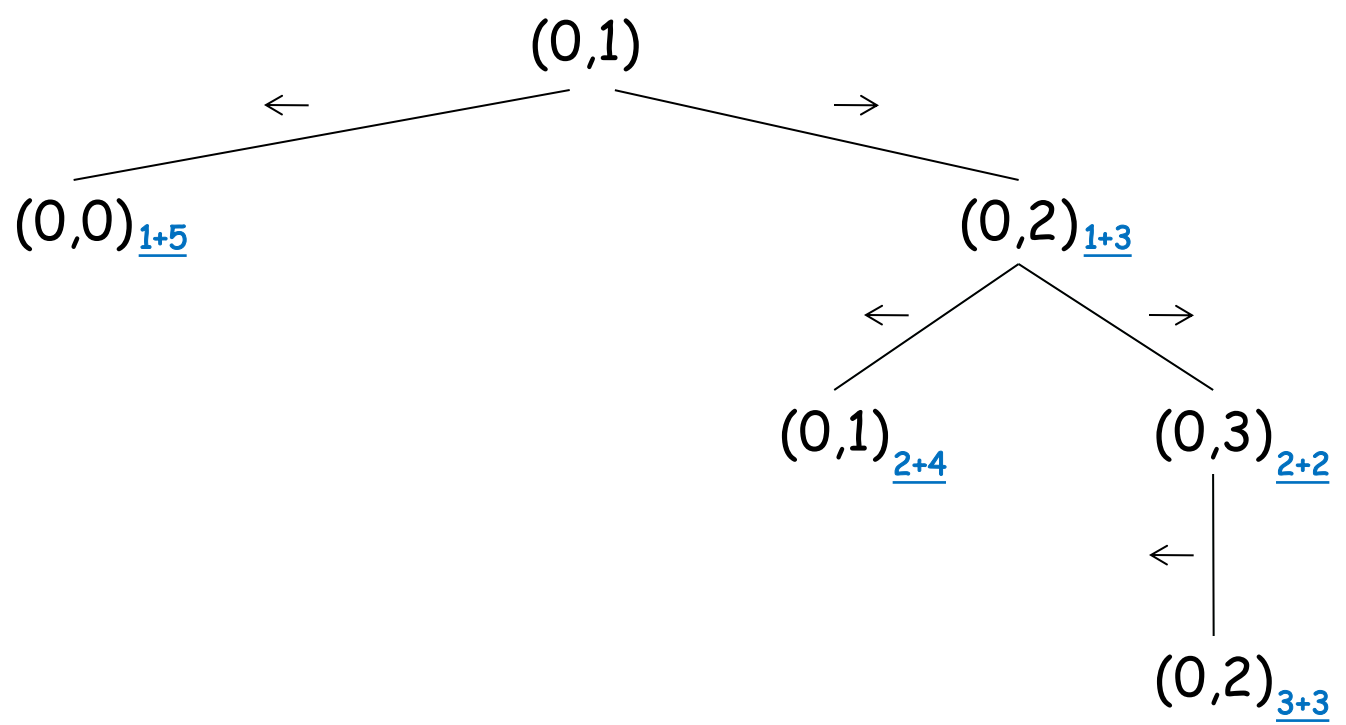
	0	1	2	3
0				
1				
2				



	0	1	2	3
0				
1				
2				

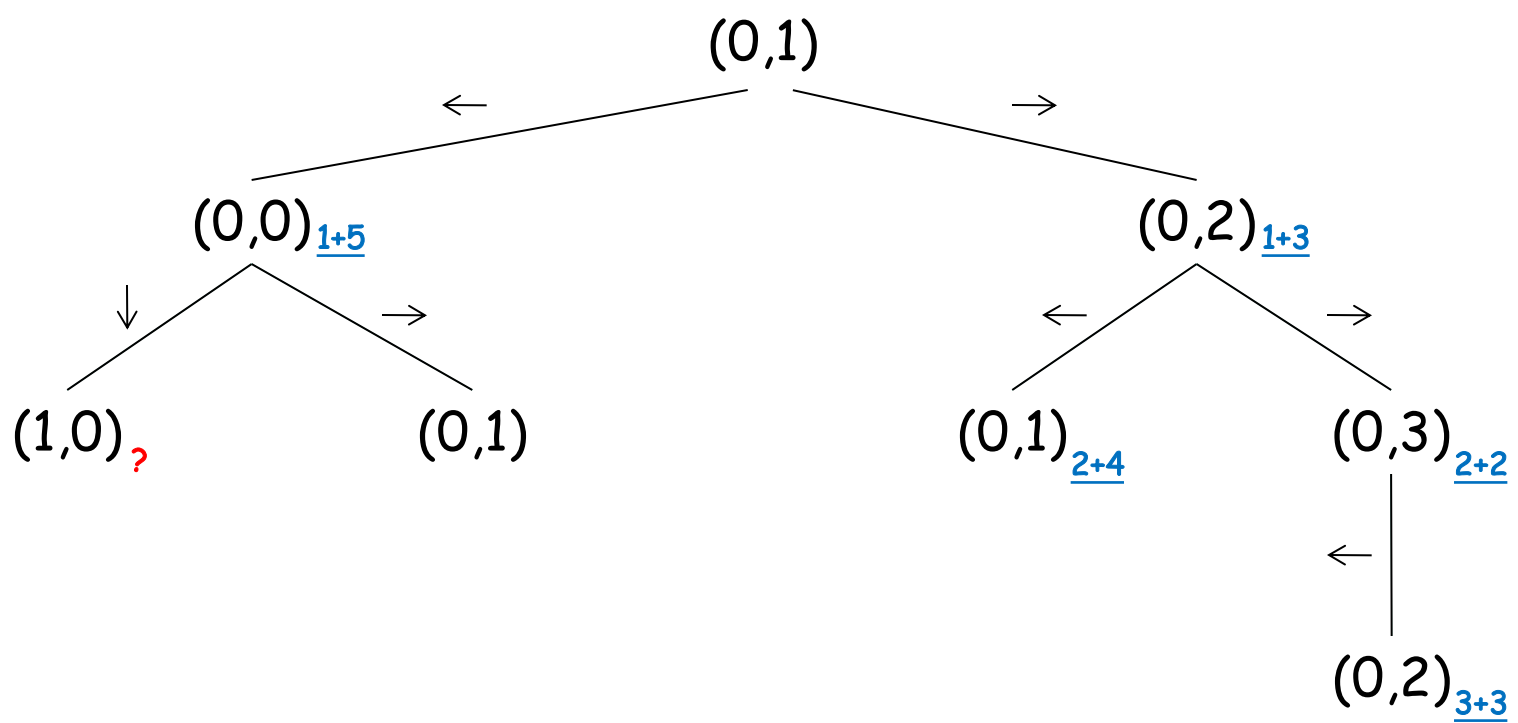


	0	1	2	3
0				
1				
2				

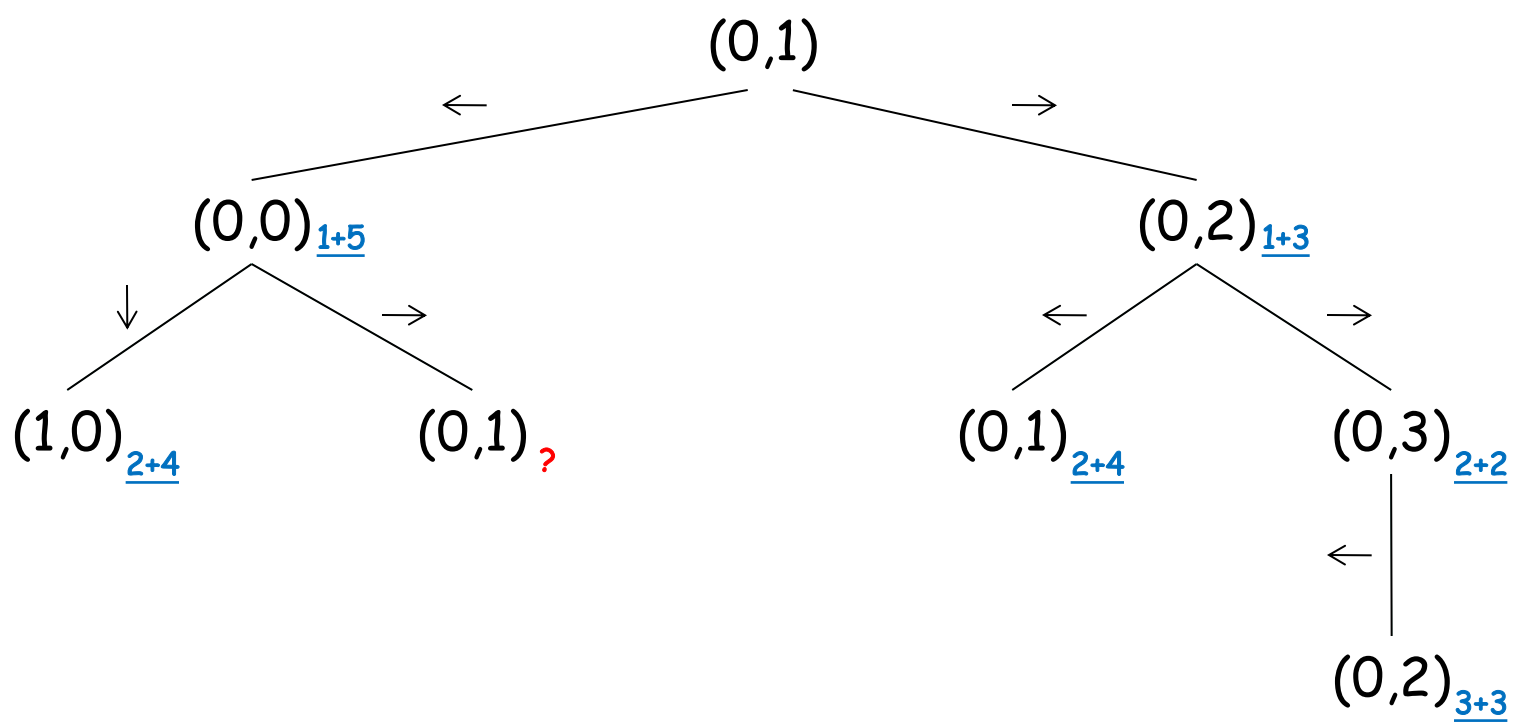


	0	1	2	3
0				
1				
2				

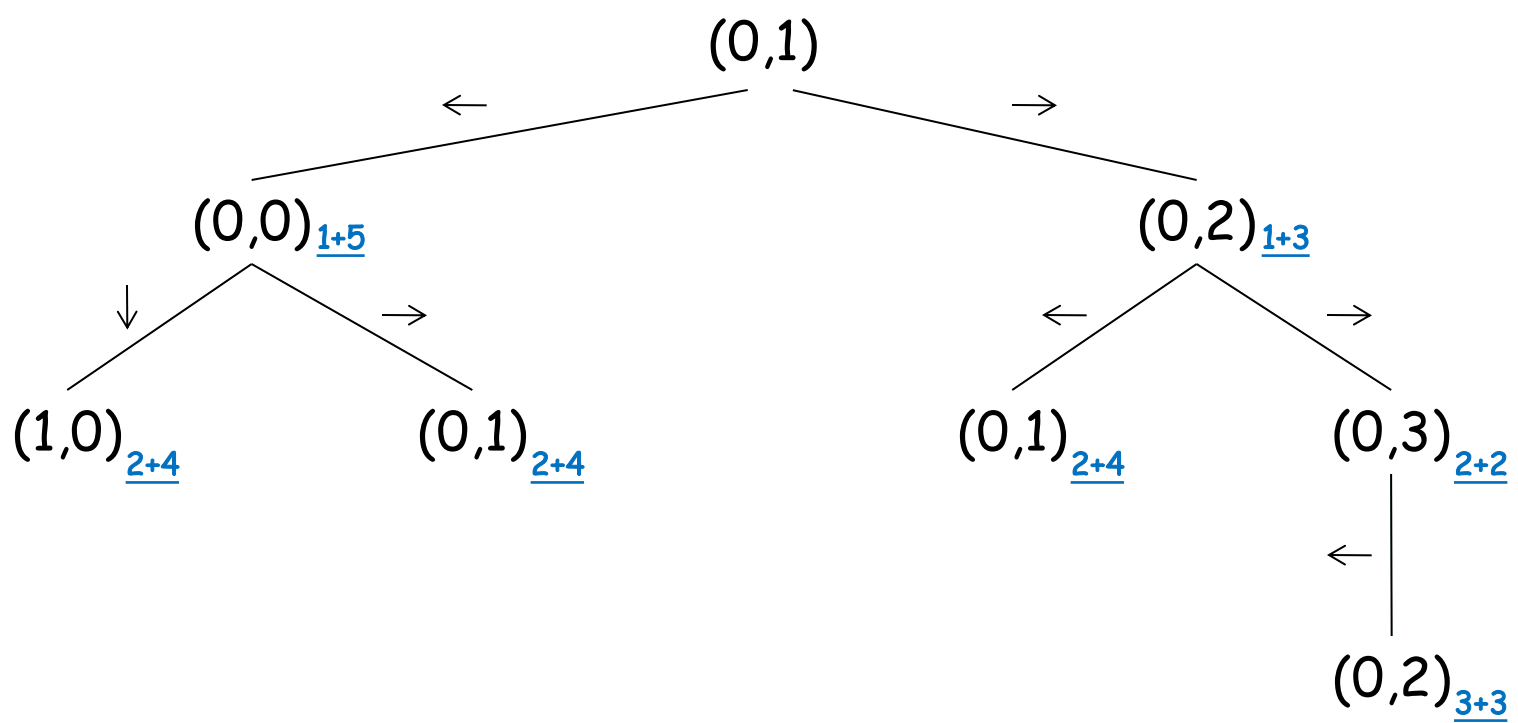




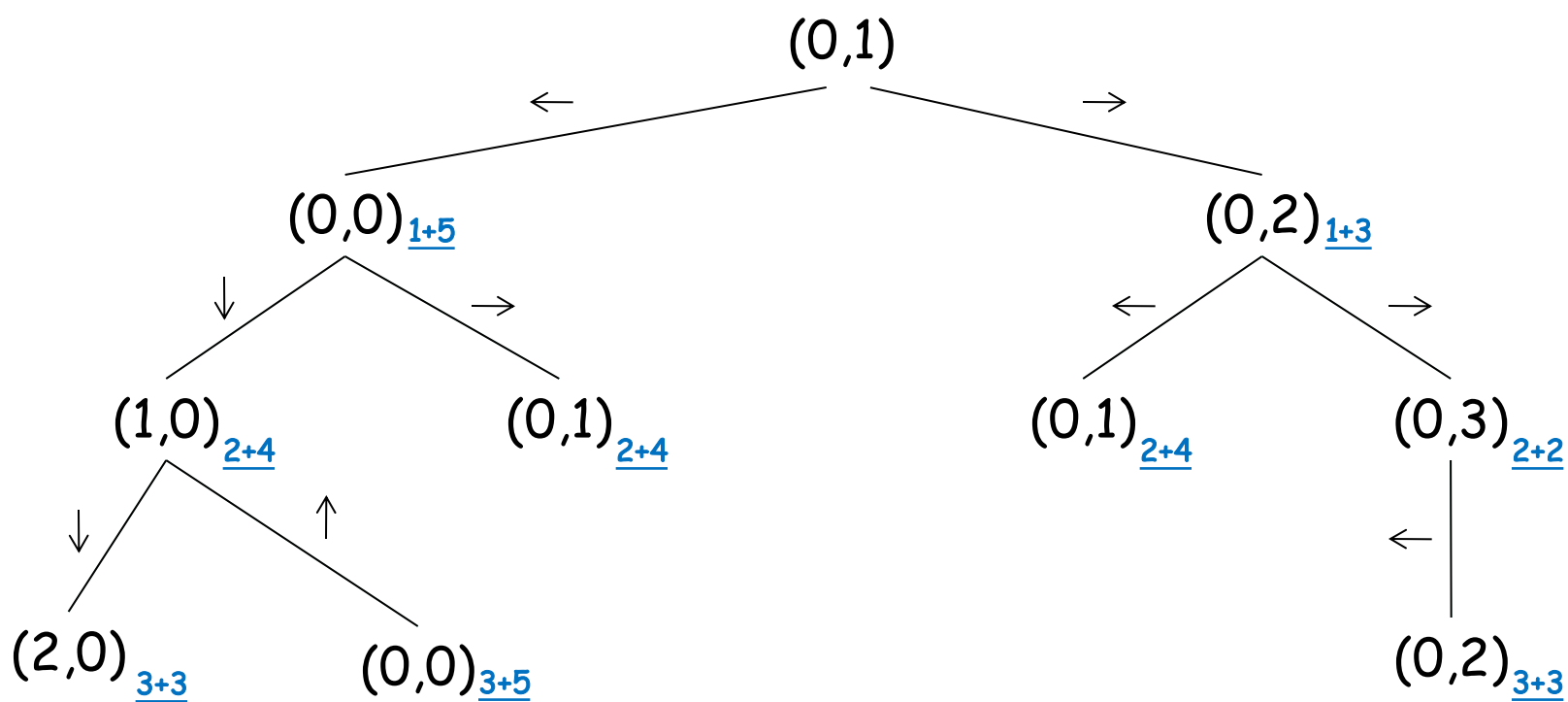
	0	1	2	3
0				
1				
2				



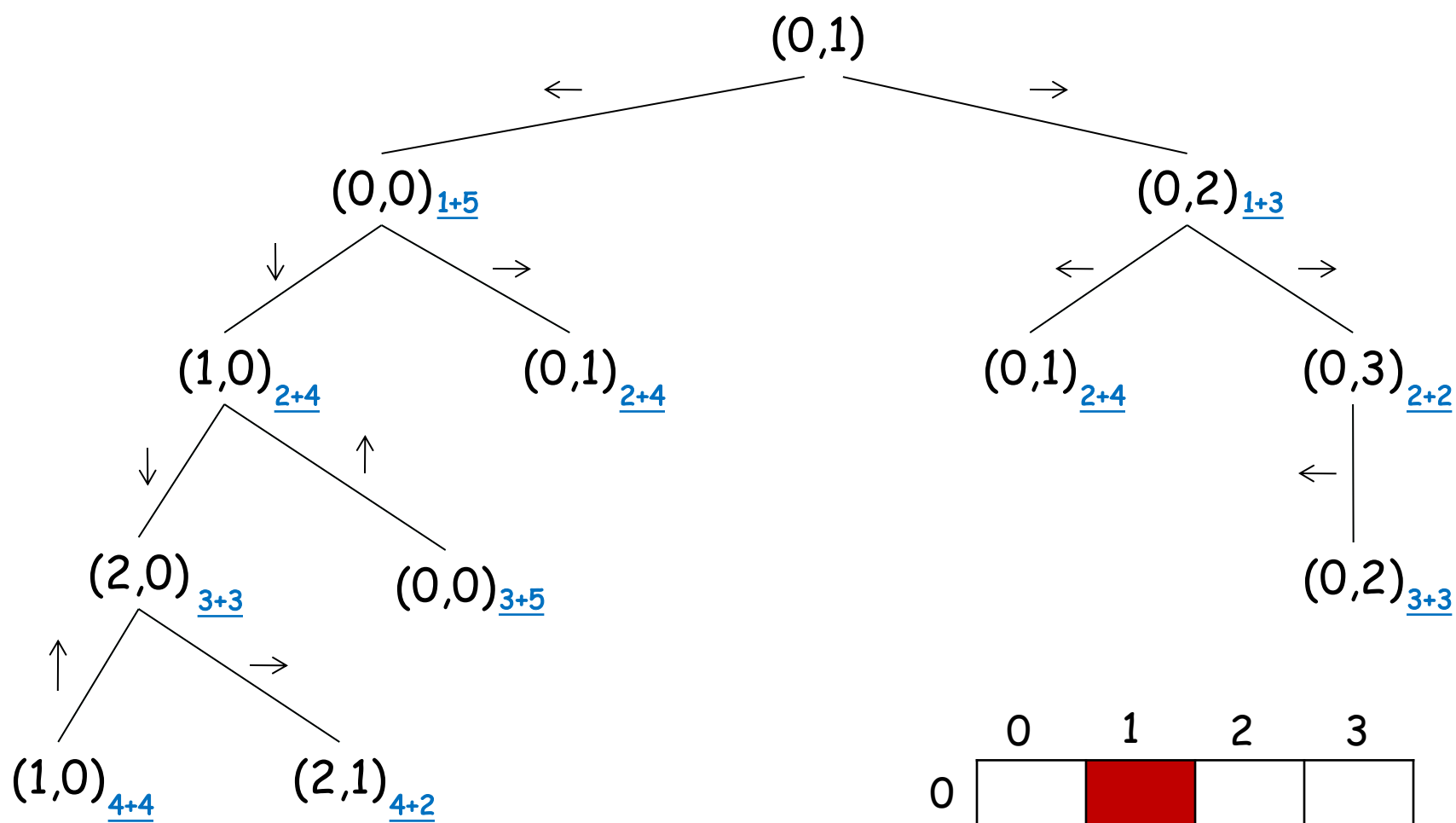
	0	1	2	3
0				
1				
2				



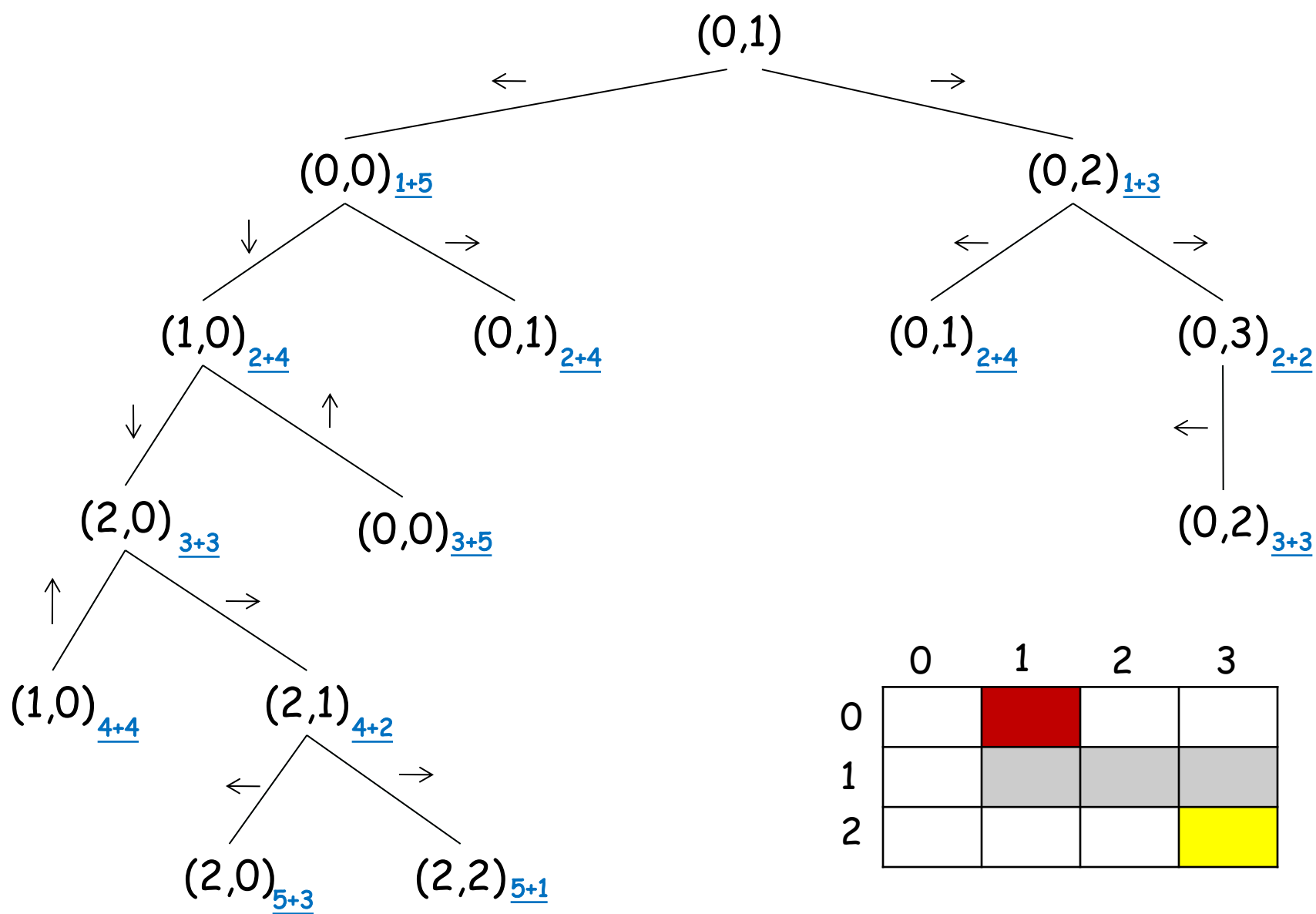
	0	1	2	3
0				
1				
2				



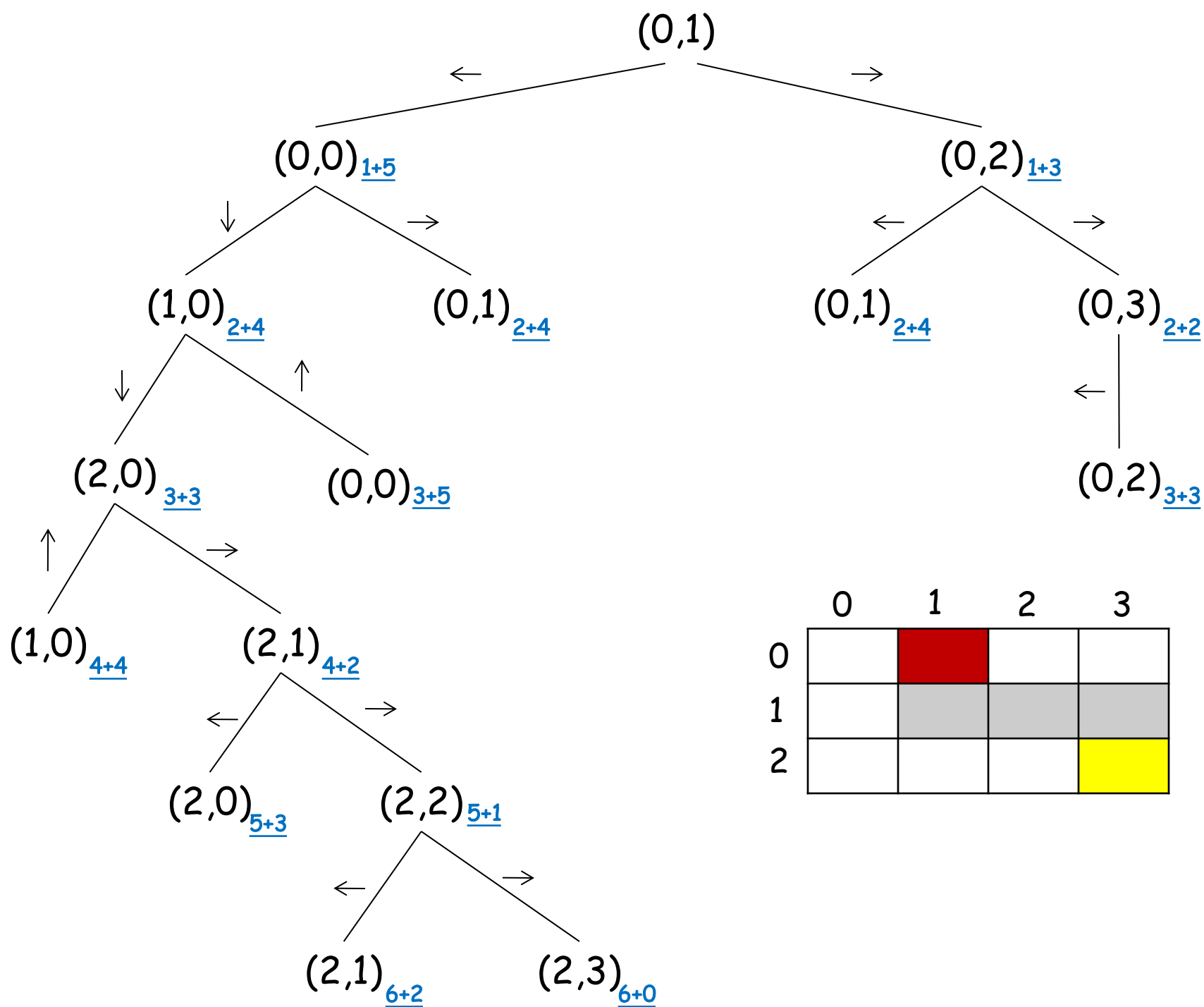
	0	1	2	3
0				
1				
2				



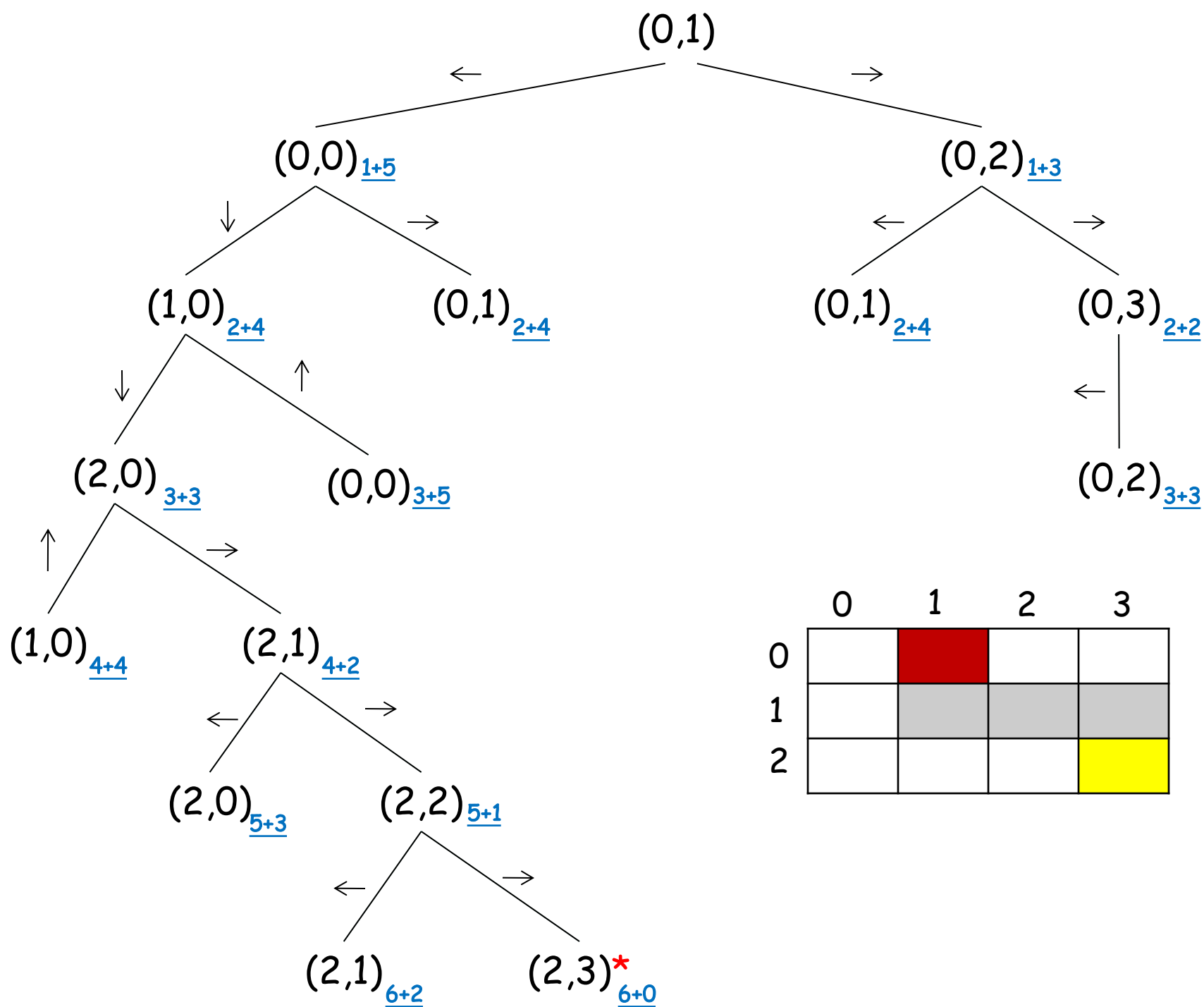
	0	1	2	3
0				
1				
2				



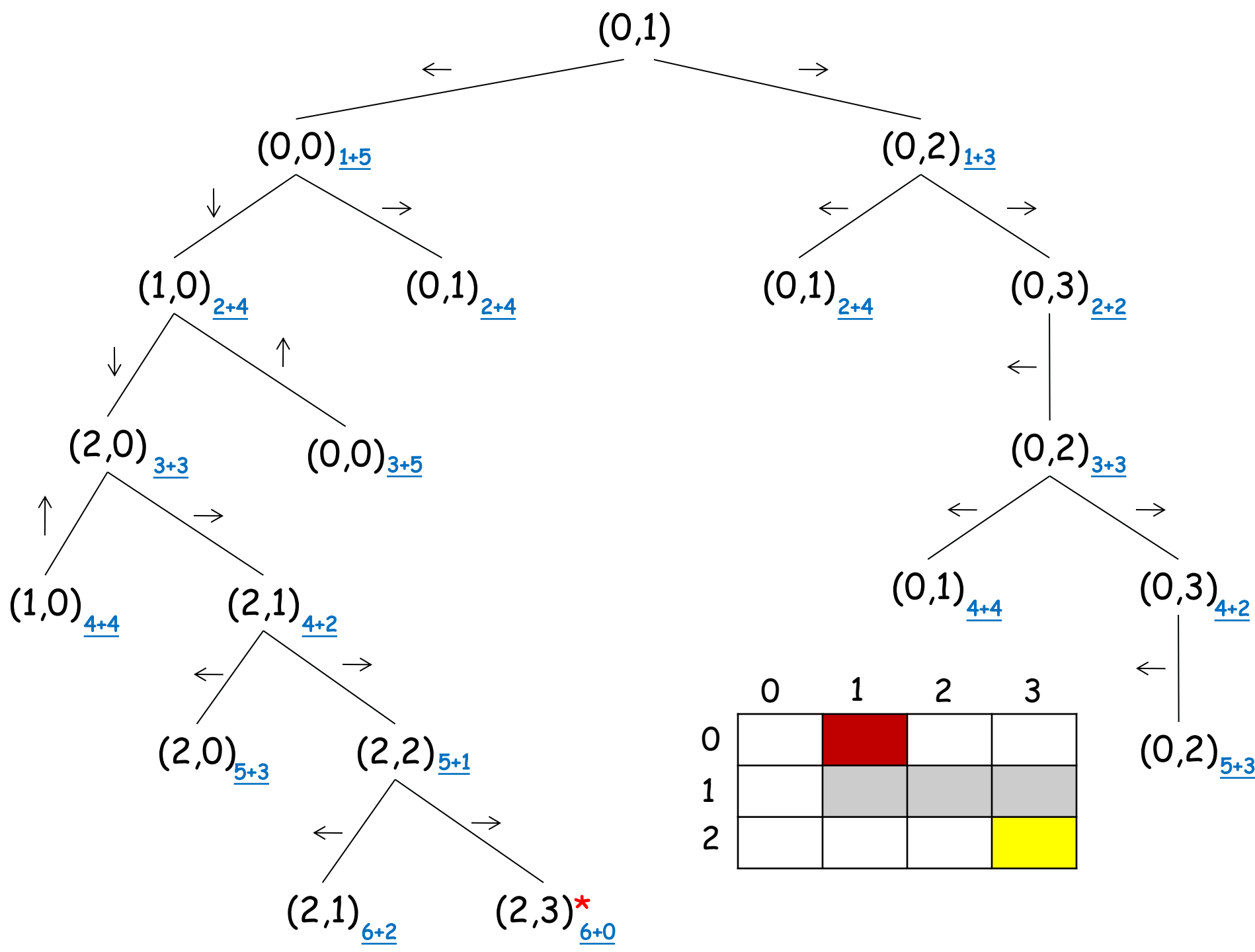
	0	1	2	3
0				
1				
2				



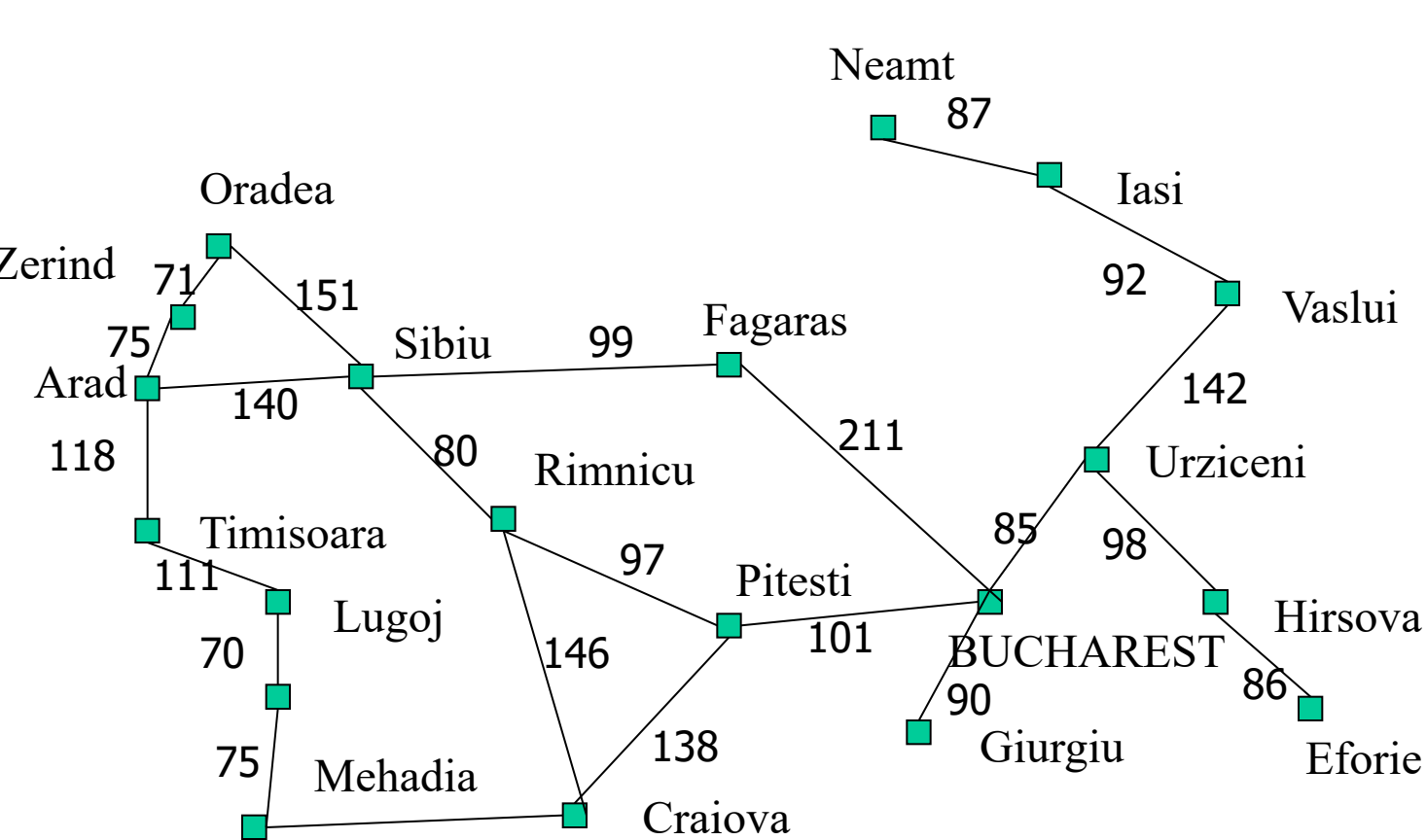
	0	1	2	3
0				
1				
2				







# Búsqueda informada



**Distancia en  
línea recta a  
Bucarest:**

Arad	366
Bucarest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

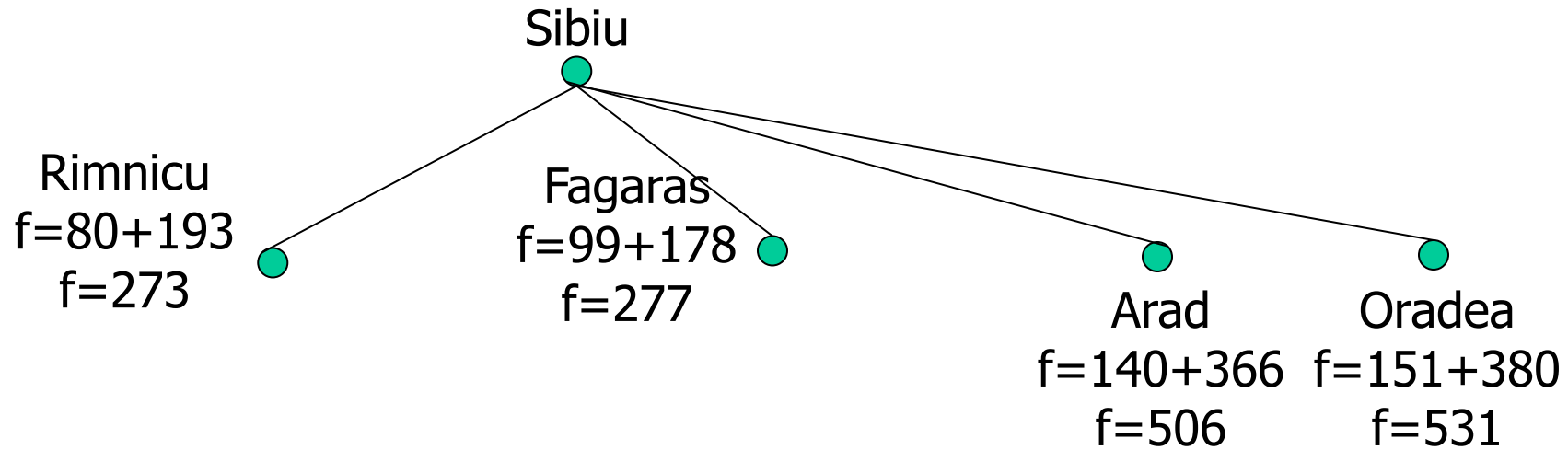
## Problema: ir de Sibiu a Bucarest

*Evite devolverse*

## Aplique $A^*$

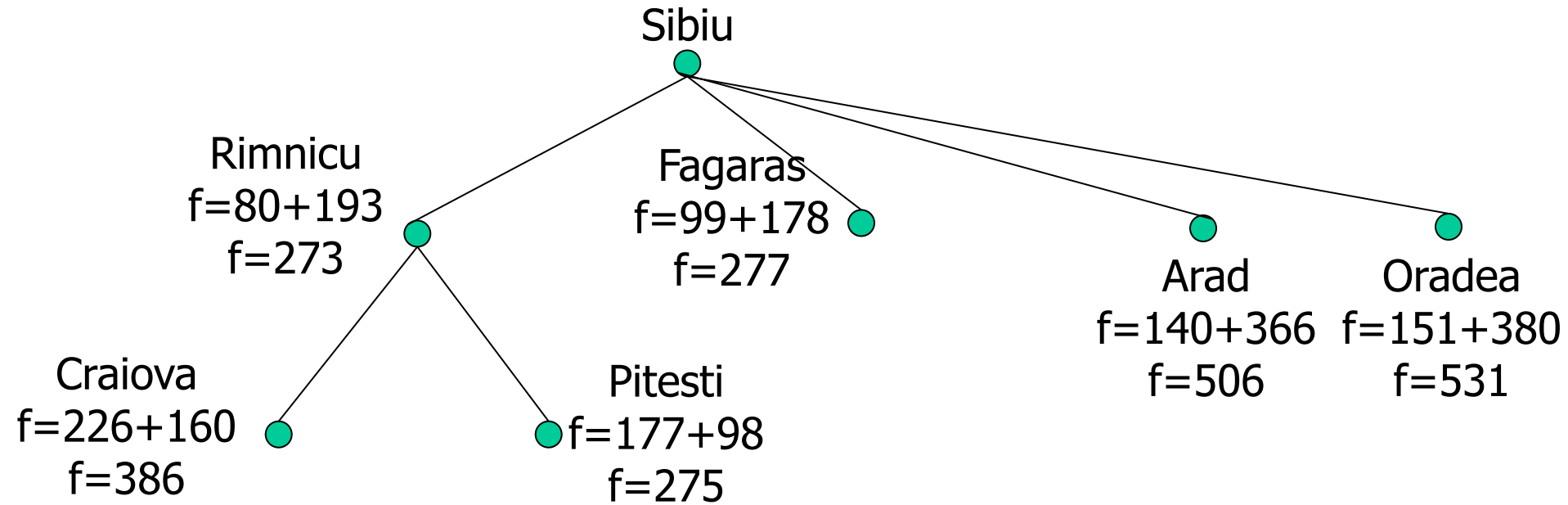
# Búsqueda informada

---



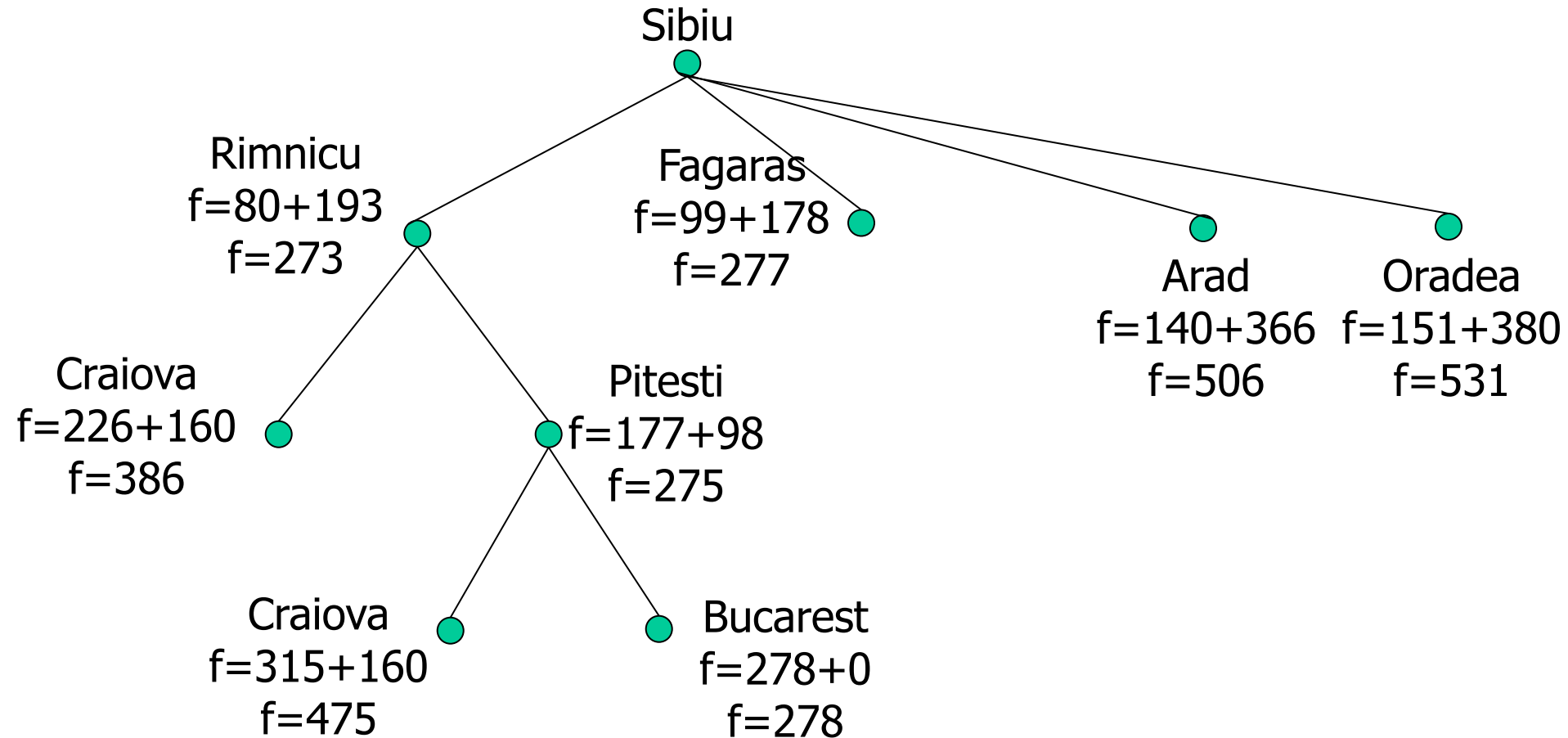
# Búsqueda informada

---



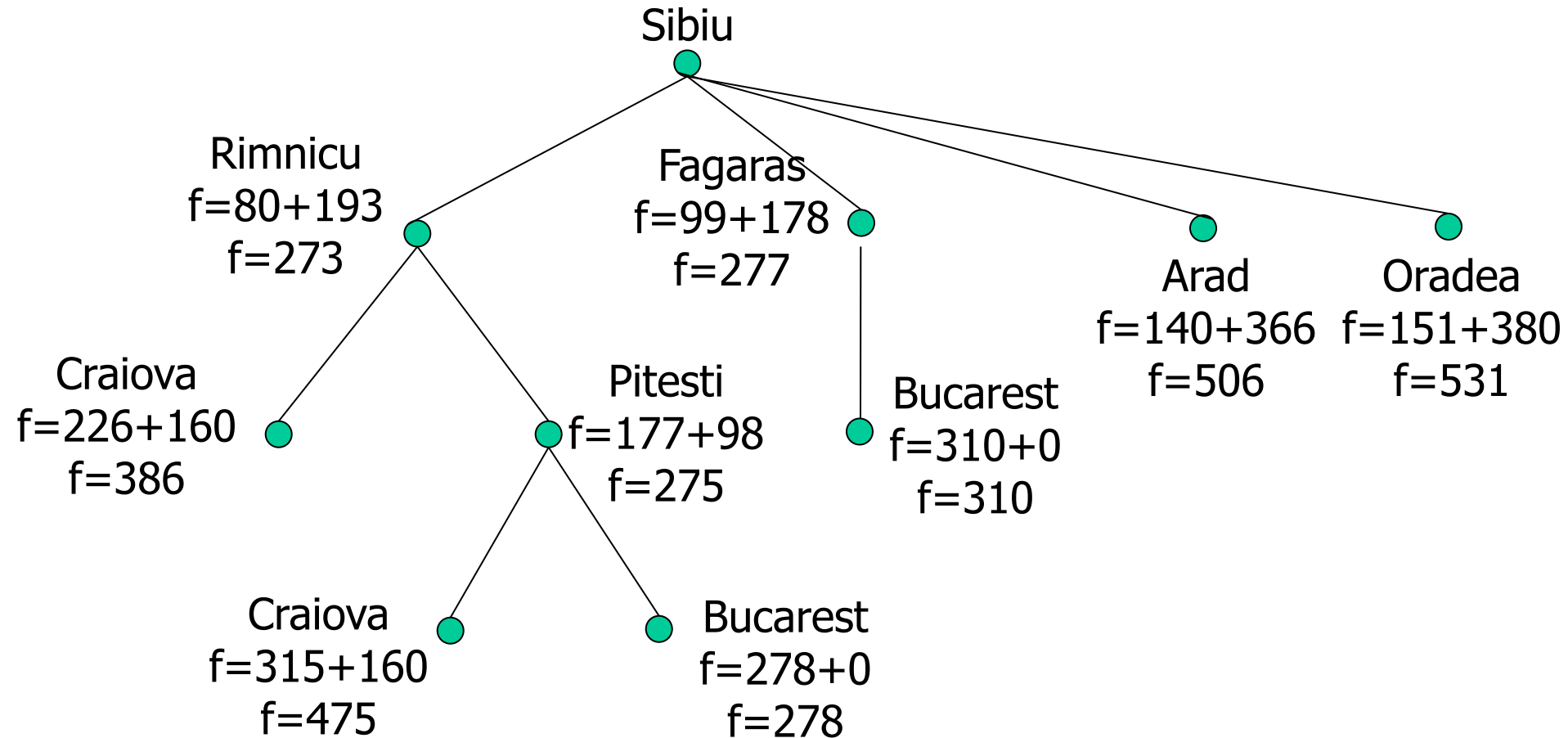
# Búsqueda informada

---



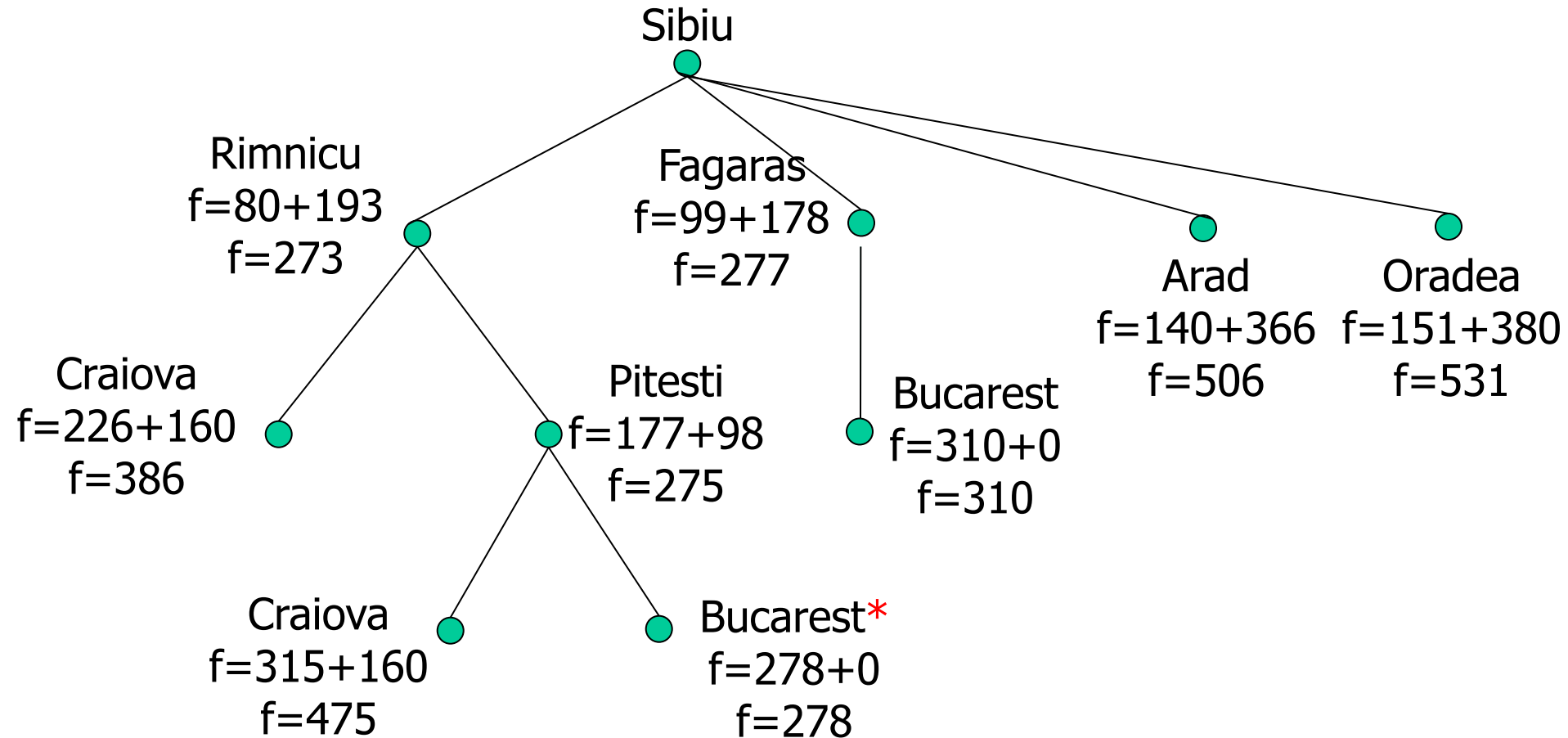
# Búsqueda informada

---

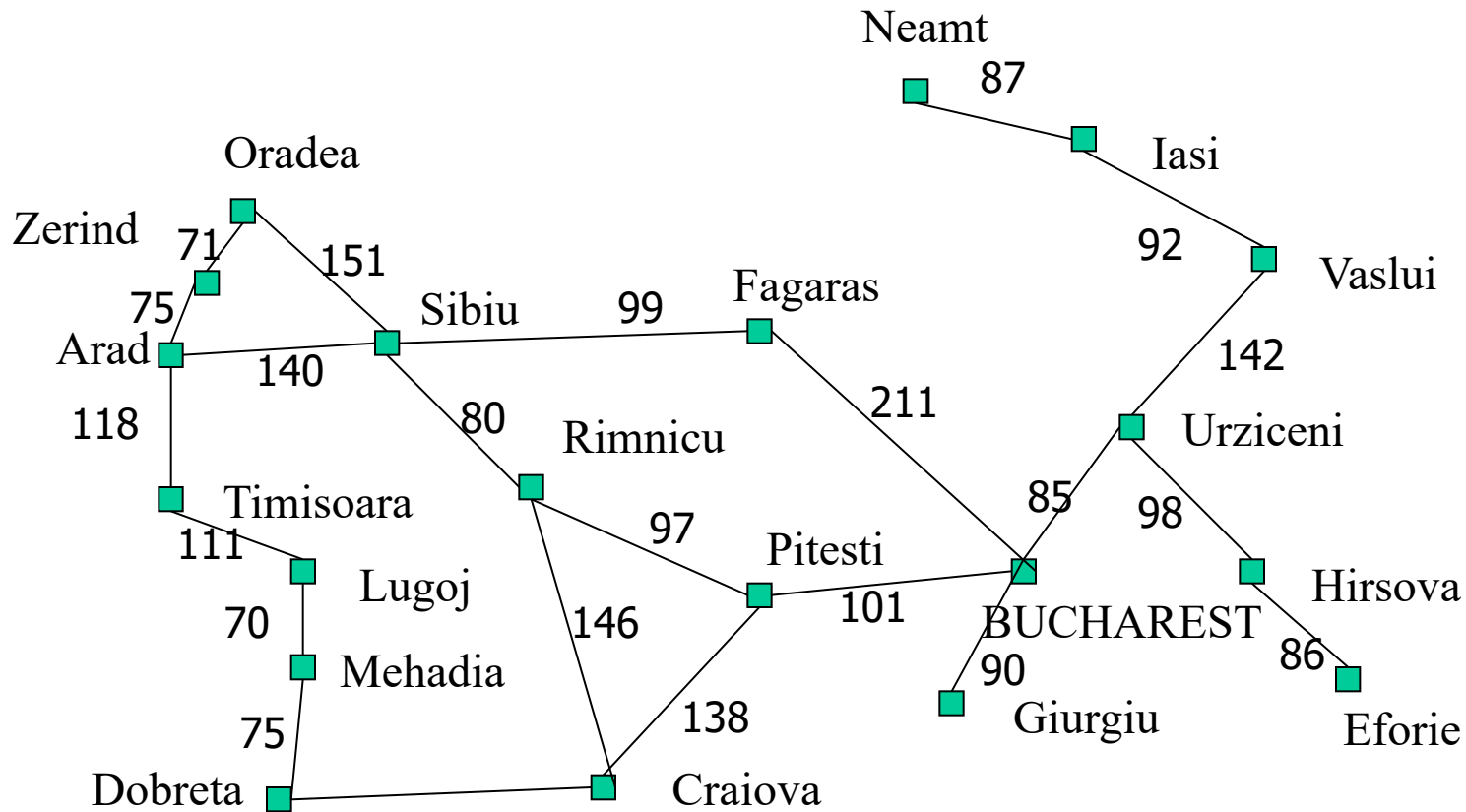


# Búsqueda informada

---



# Búsqueda informada



La solución obtenida es  
*Sibiu* → *Rimnicu* → *Pitesti* → *Bucarest* que tiene  
costo **278** y es la óptima



# Búsqueda informada









## Aplicar A\*

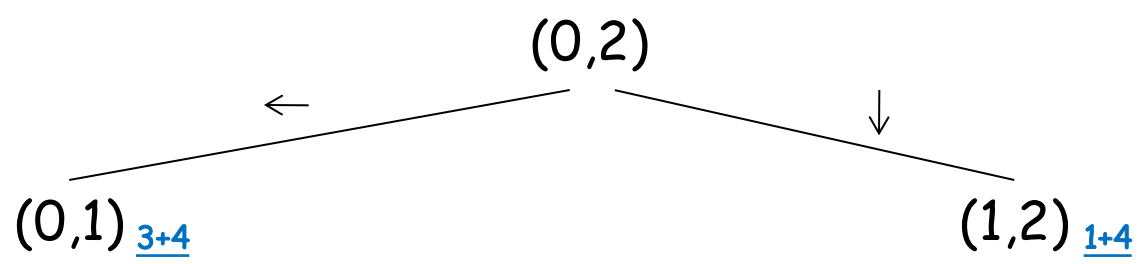
### El ratón inteligente.









- Llegar a una casilla libre cuesta 1
- Llegar a una casilla donde hay gato cuesta 5
- Llegar a una casilla donde hay ratonera cuesta 3

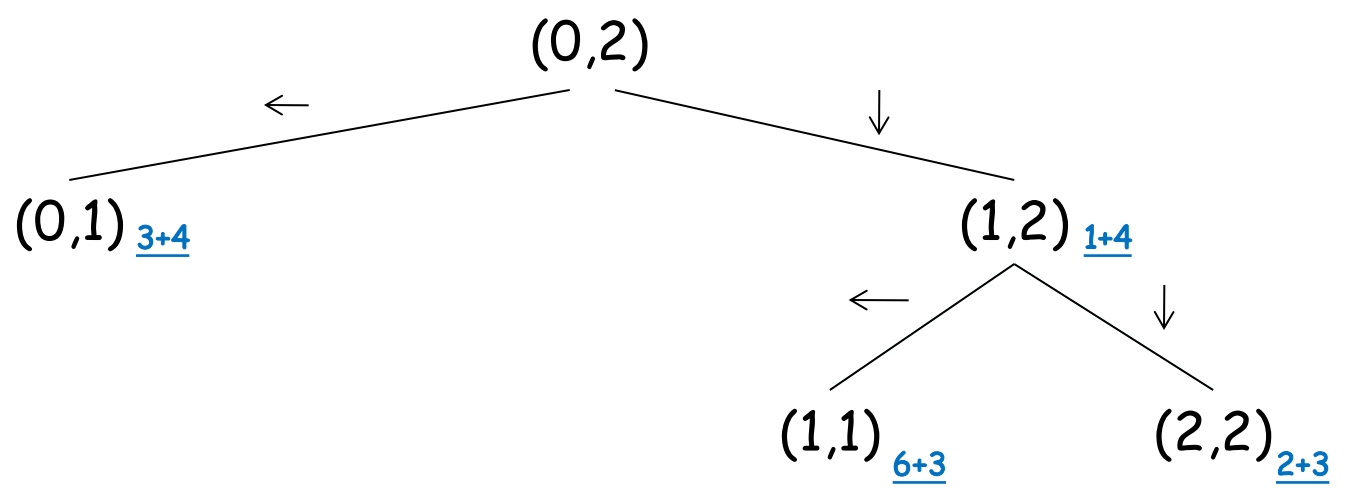
*Evite devolverse*









*Compare el resultado con Avara*

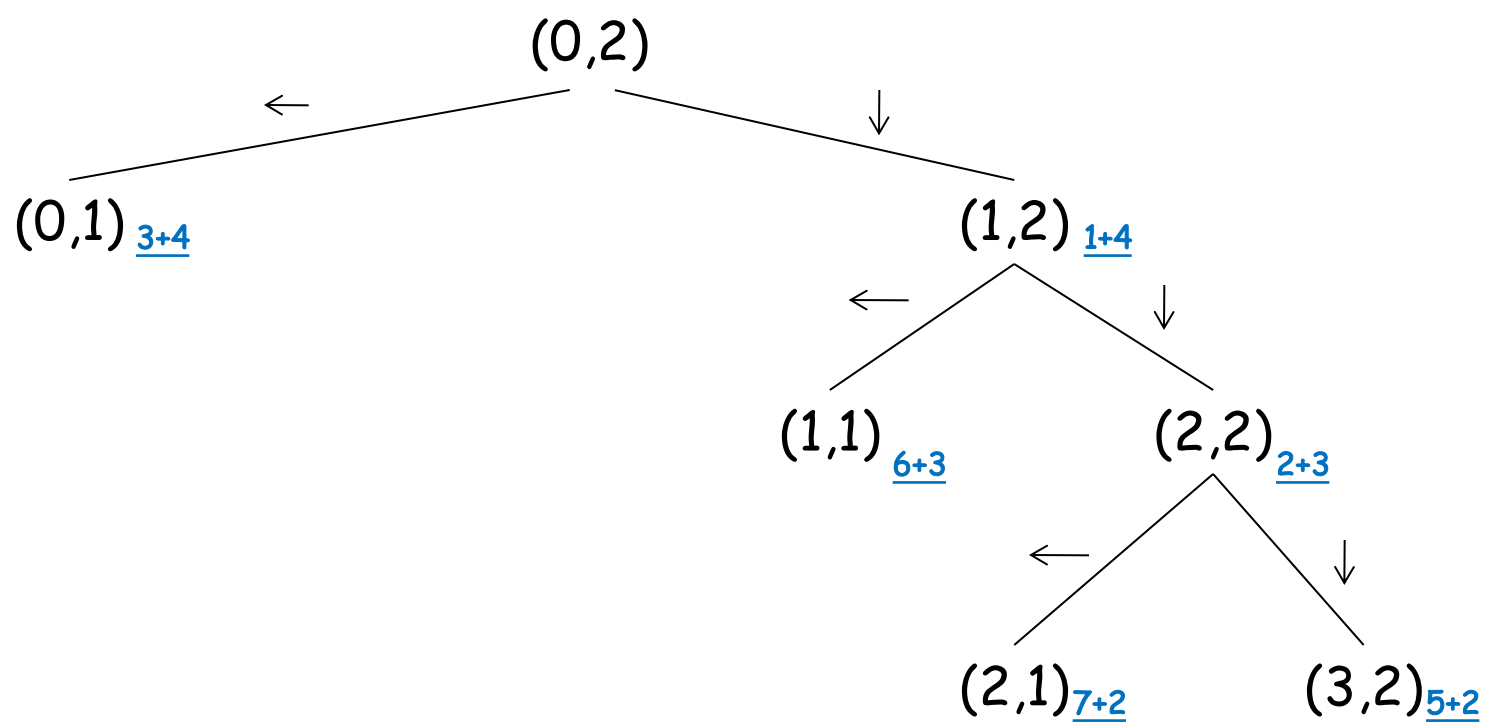
	0	1	2
0			
1			
2			
3			
4			











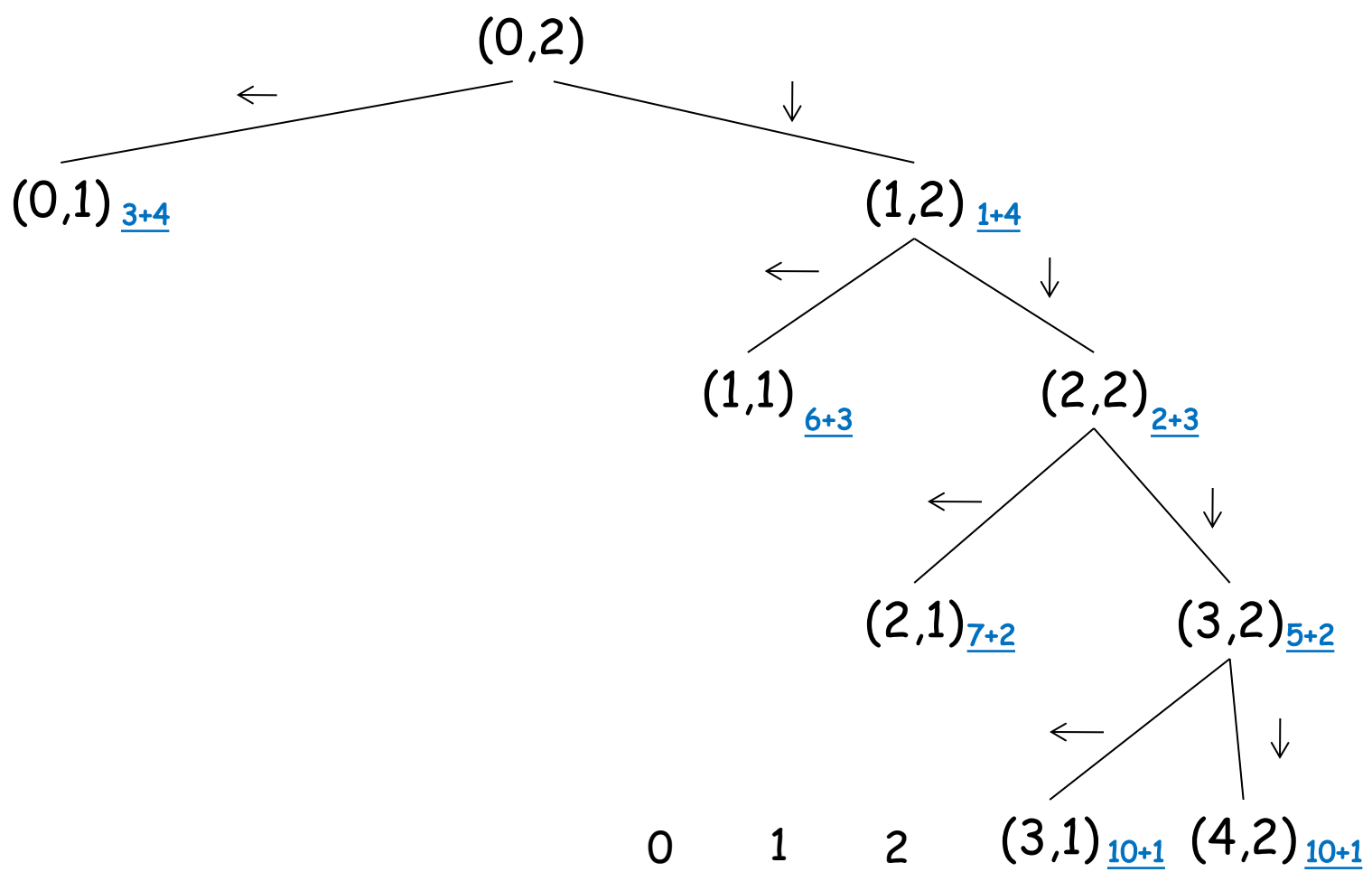
	0	1	2
0			
1			
2			
3			
4			





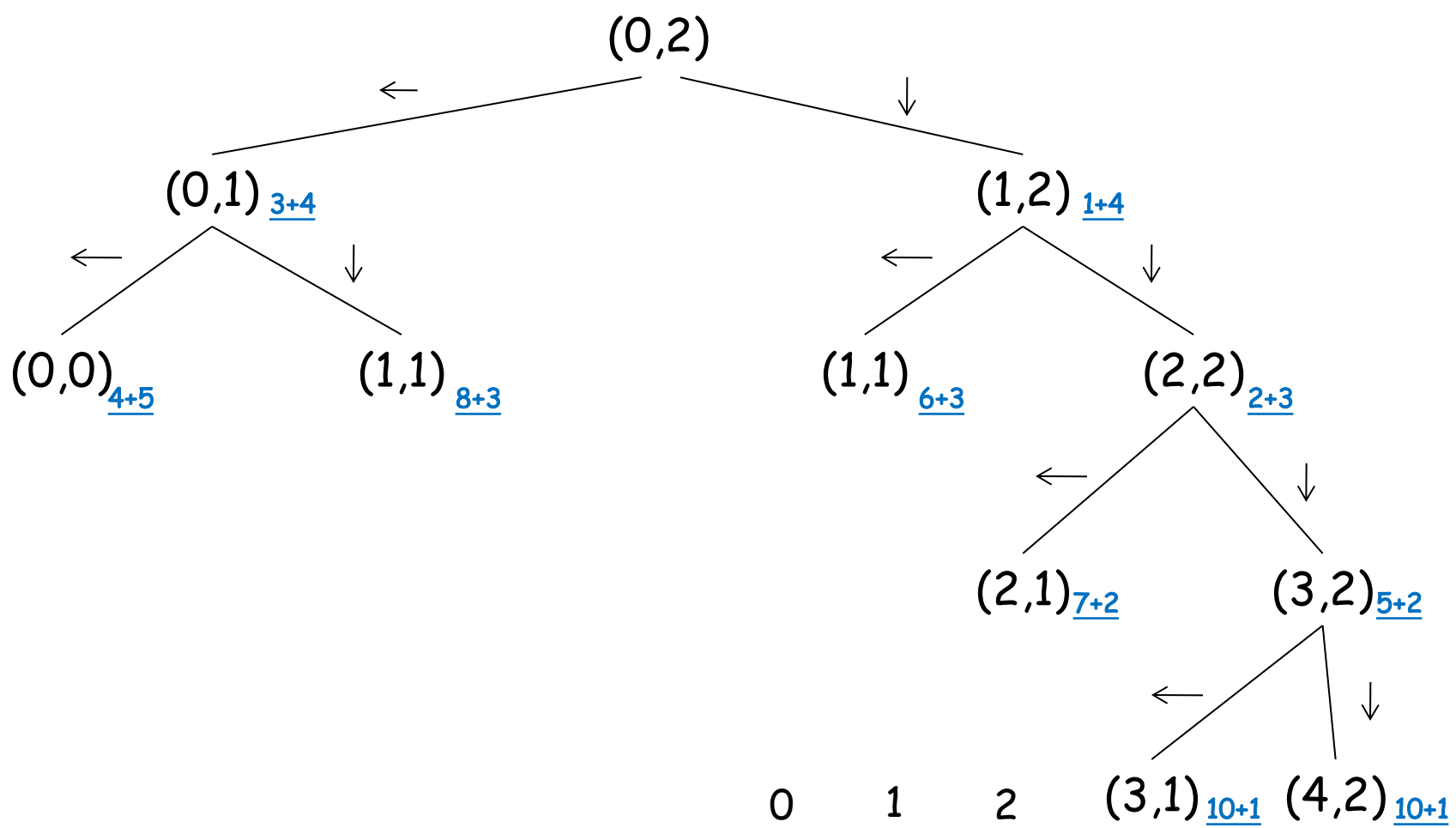
	0	1	2
0			
1			
2			
3			
4			











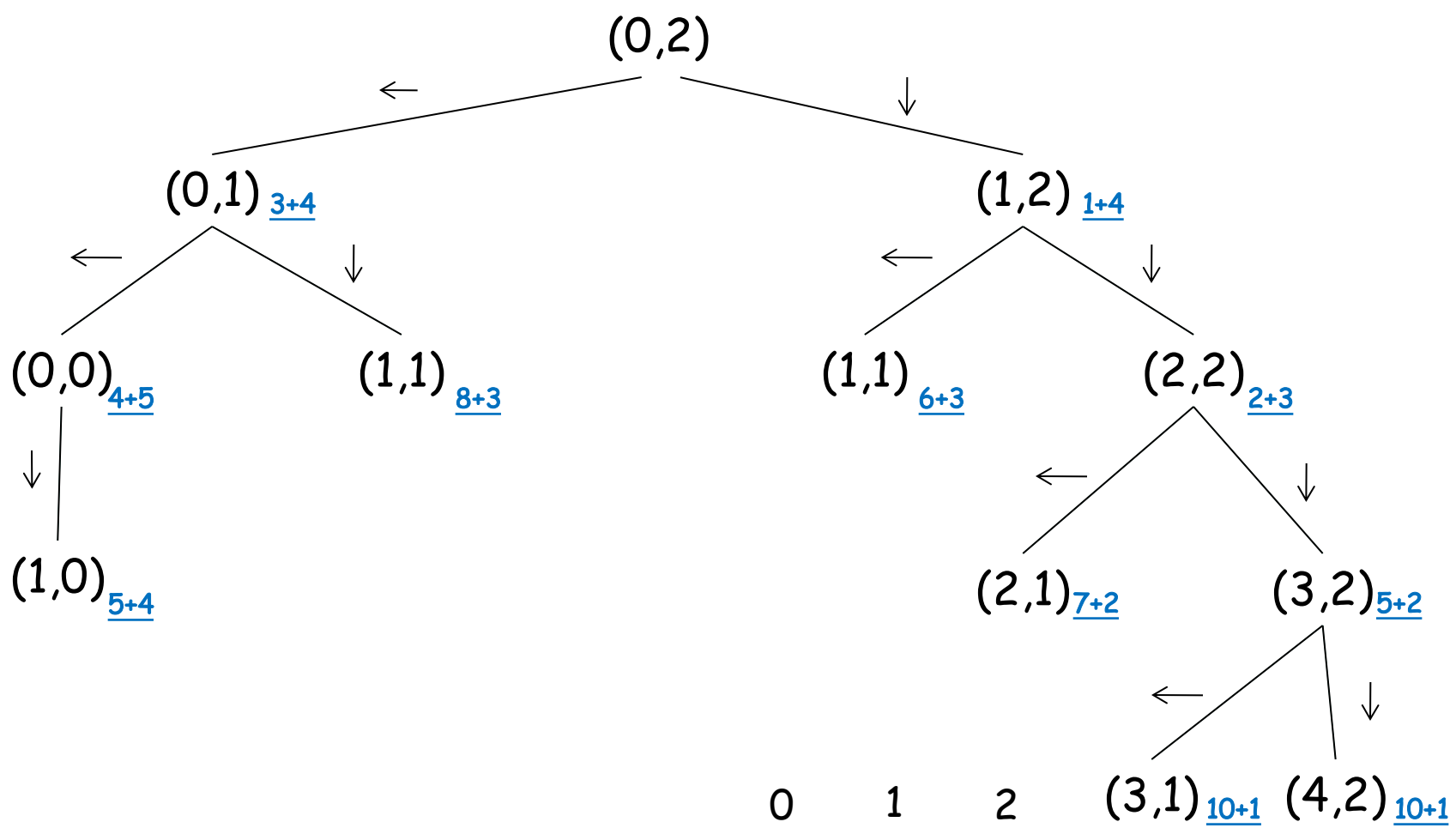
	0	1	2
0			
1			
2			
3			
4			











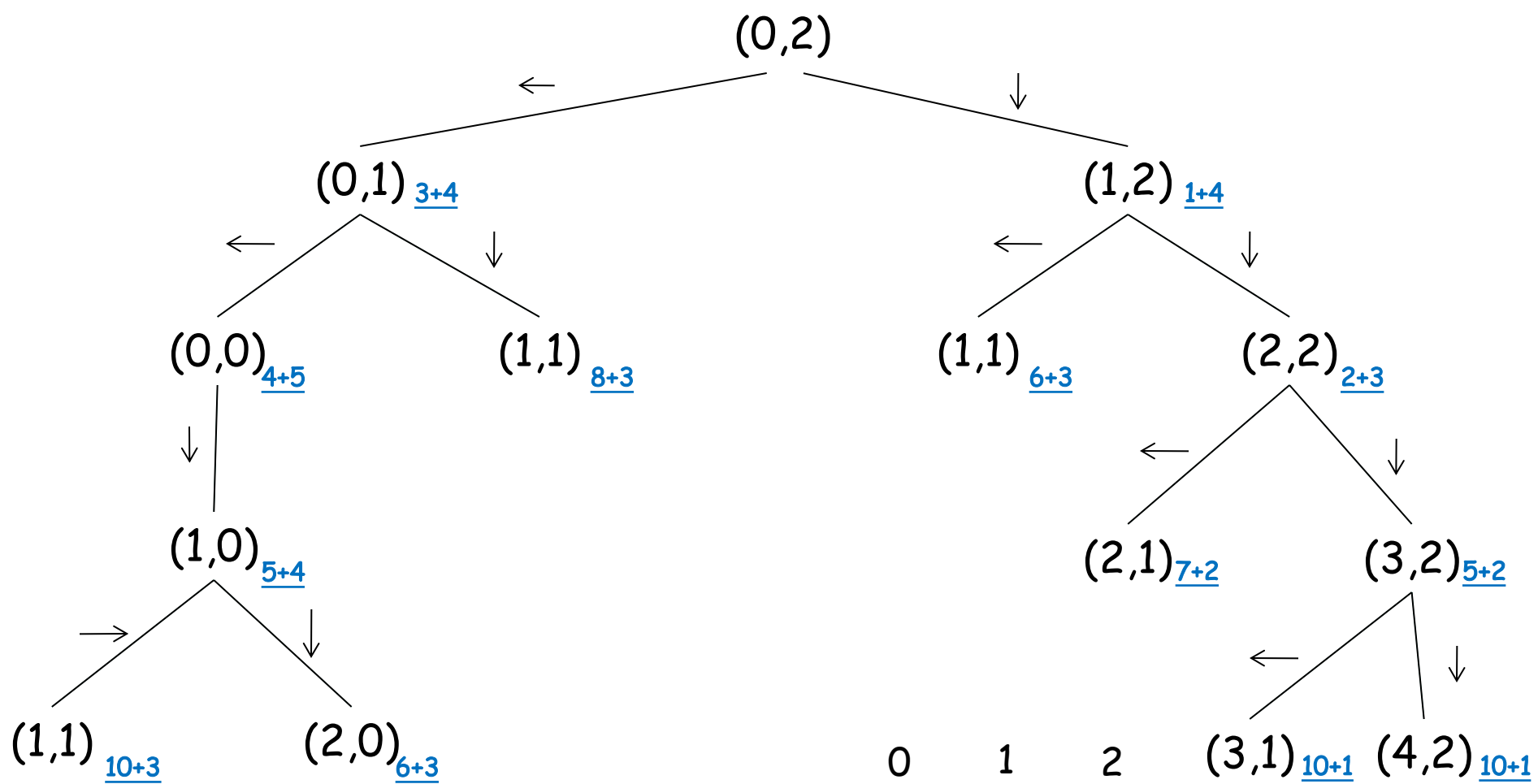
	0	1	2
0			
1			
2			
3			
4			











	0	1	2
0			
1			
2			
3			
4			

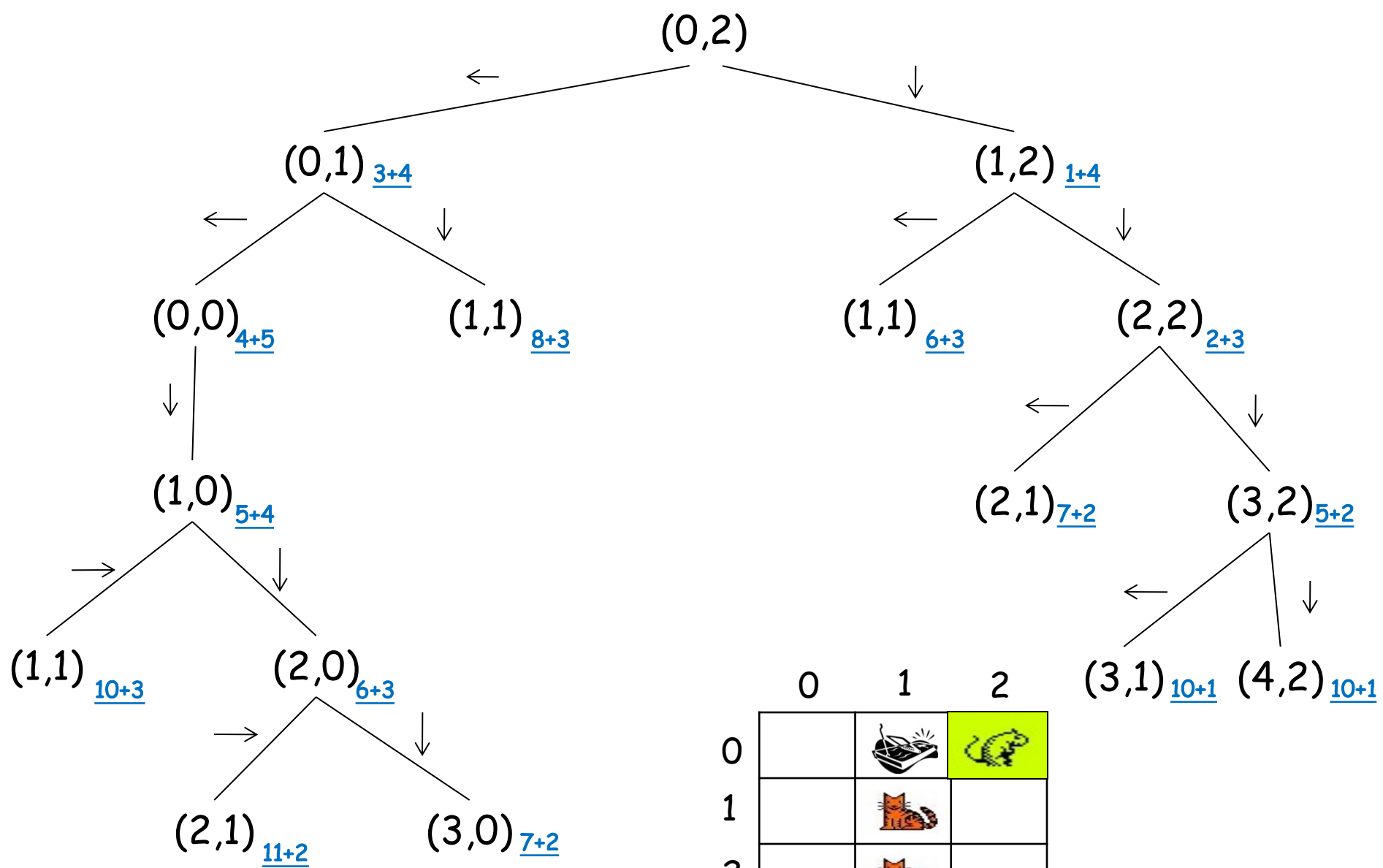










	0	1	2
0			
1			
2			
3			
4			

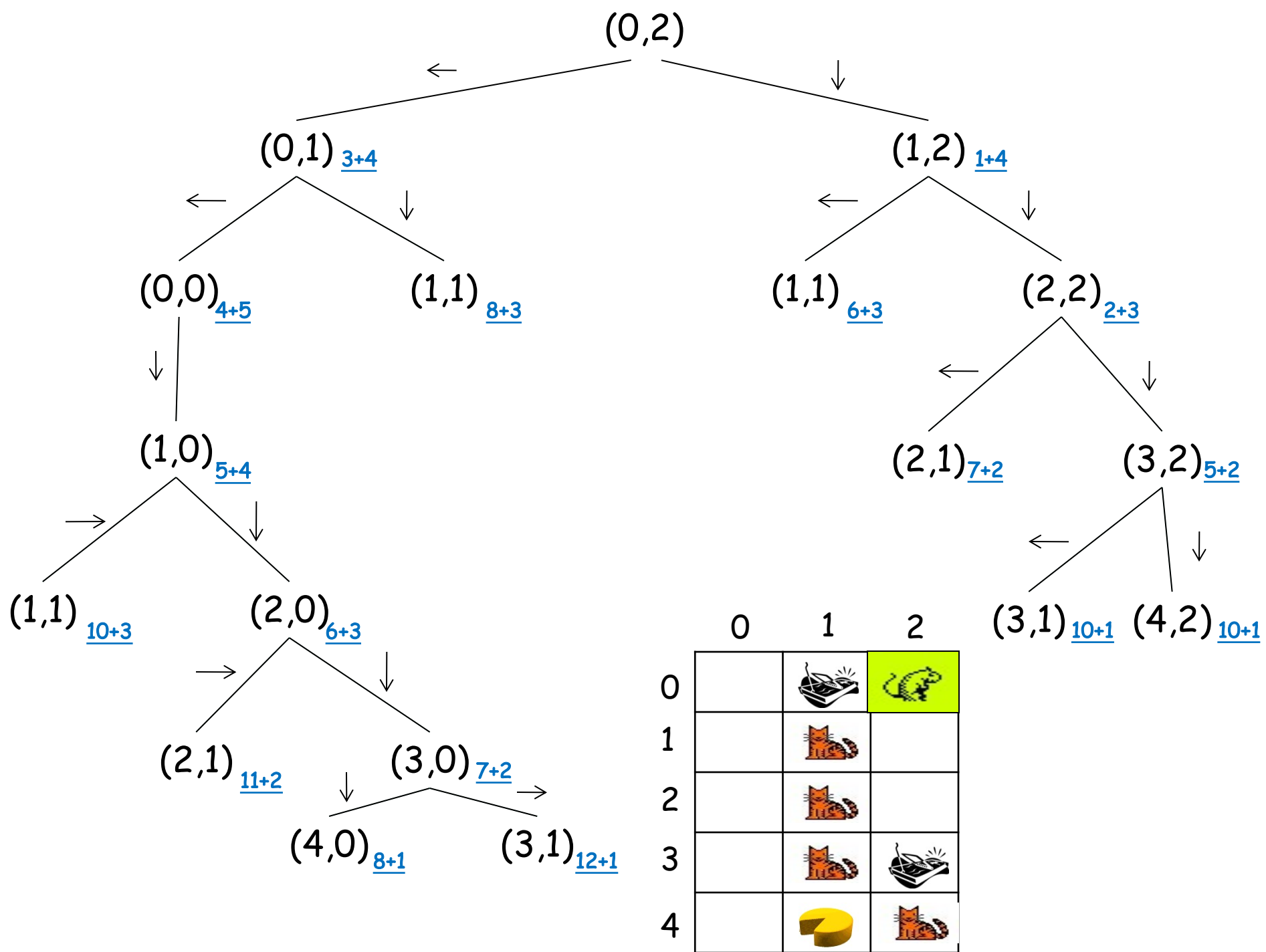


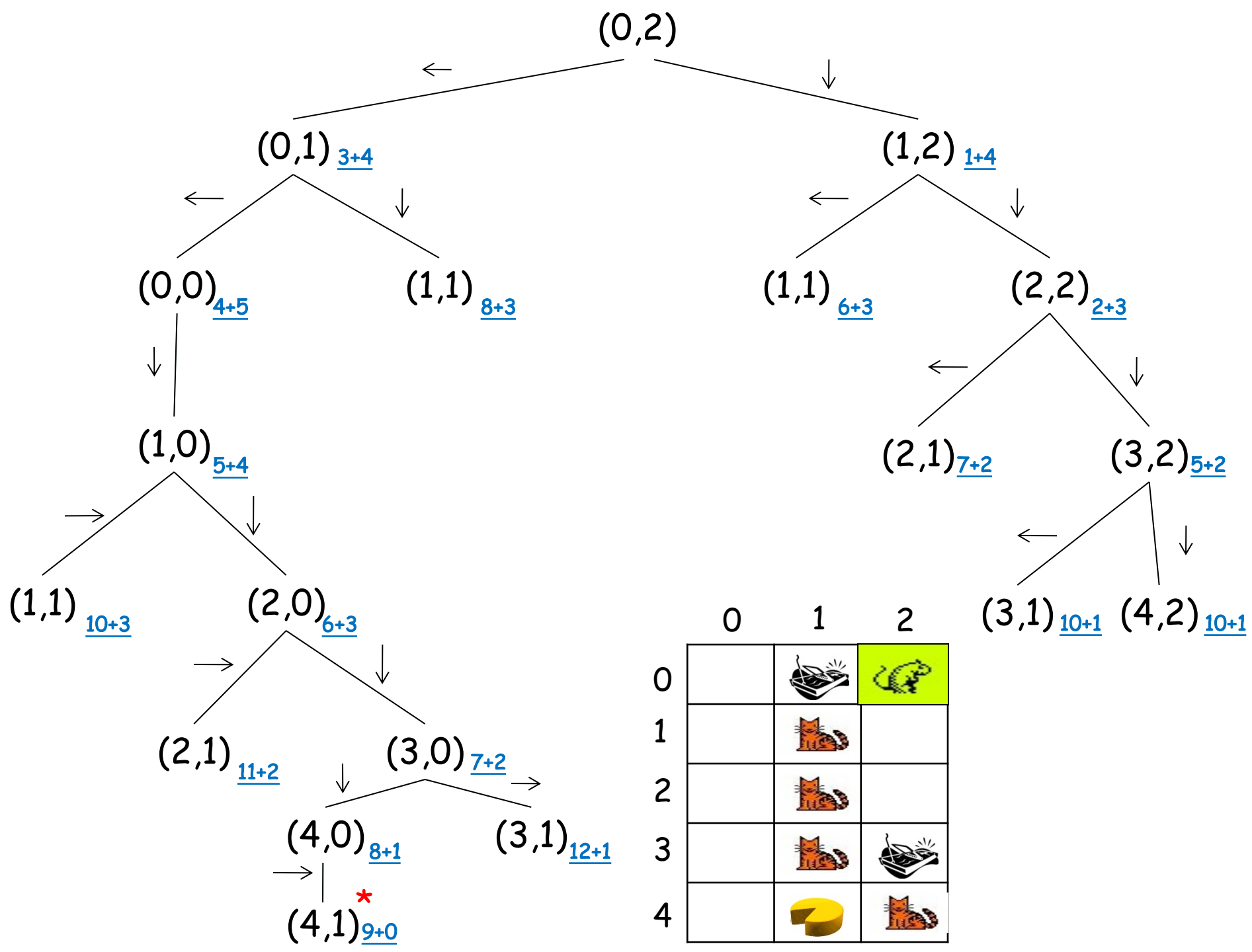
	0	1	2
0			
1			
2			
3			
4			





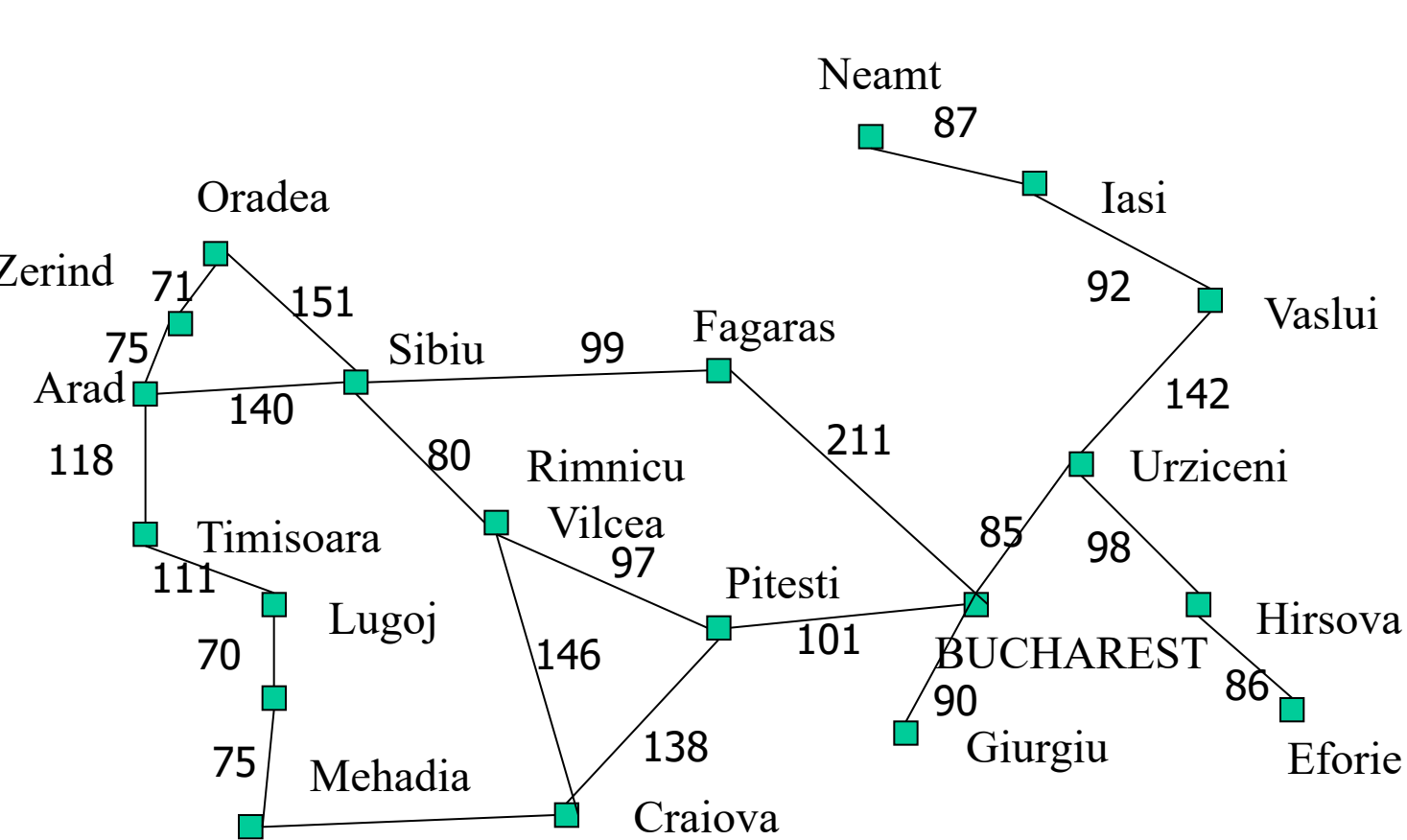
	0	1	2
0			
1			
2			
3			
4			





# Búsqueda informada

Aplique A\* para ir de Iasi a Fagaras



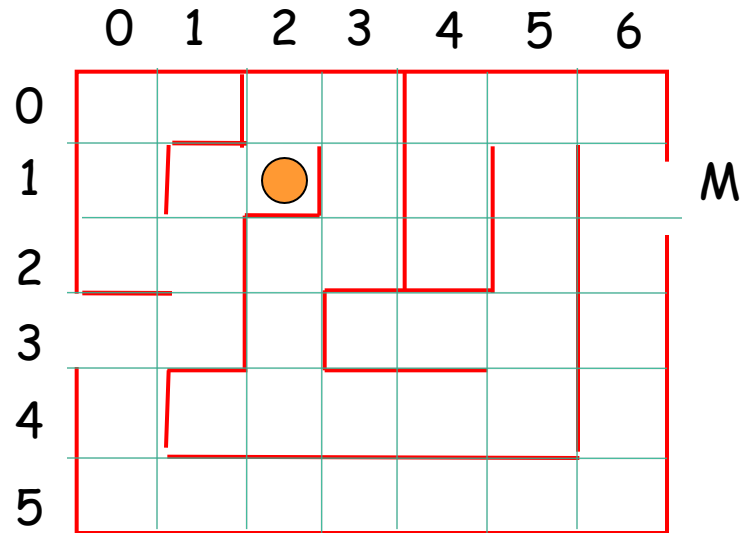
Distancia en  
línea recta a  
Fagaras:

Bucarest	185
Fagaras	0
Giurgiu	182
Iasi	160
Hirsova	190
Pitesti	170
Neamt	150
Urziceni	178
Vaslui	170

# Búsqueda informada

---

- Aplicar  $A^*$



# Búsqueda informada

---

## Implementación

- Se puede implementar considerando la lista de nodos a expandir como una cola de prioridad, donde la prioridad es el valor de  $f(n)$  y se selecciona aquel con menor prioridad

# Búsqueda informada

---

## Algoritmo $A^*$

- La estrategia es completa y óptima si se cumple que la heurística  $h(n)$  sea admisible

# Búsqueda informada

---

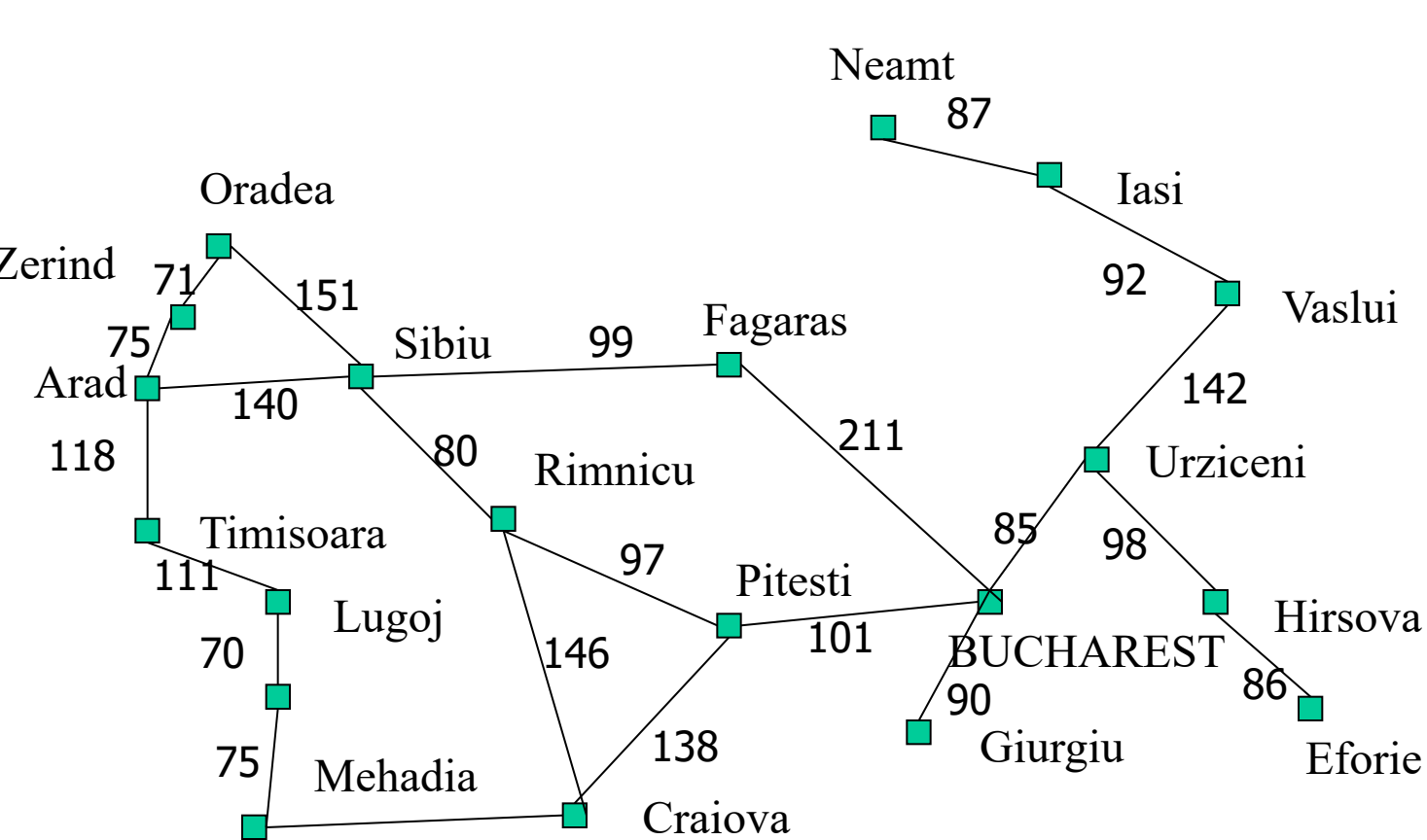
- Una **heurística es admisible** si nunca asigna un valor mayor al real, es decir, se cumple que

$$h(n) \leq \text{costo\_real}(n)$$

- Las heurísticas admisibles se conocen como **heurísticas optimistas**



# Búsqueda informada

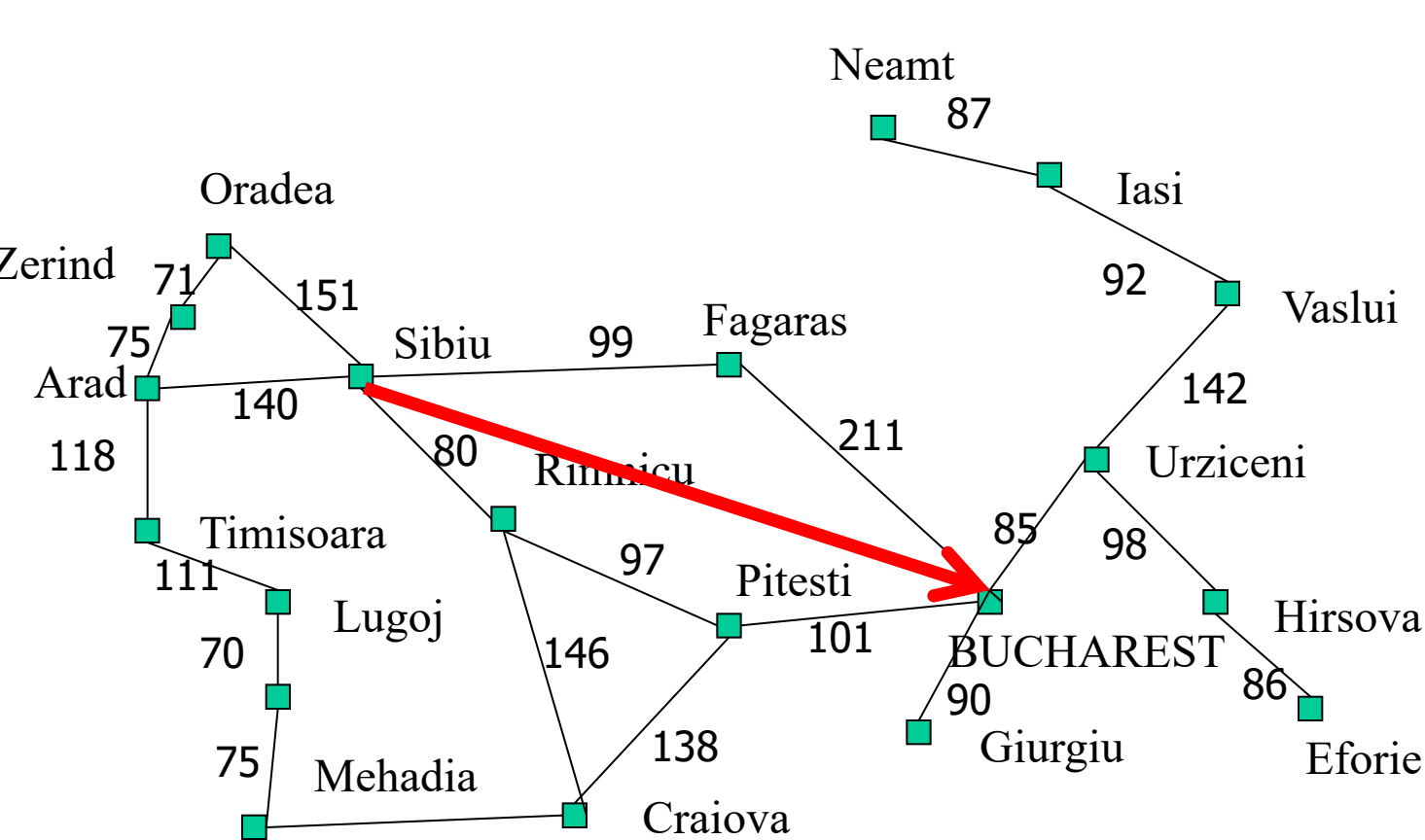


**Distancia en  
línea recta a  
Bucarest:**

Arad	366
Bucarest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

¿La distancia en línea recta es admisible?

# Búsqueda informada

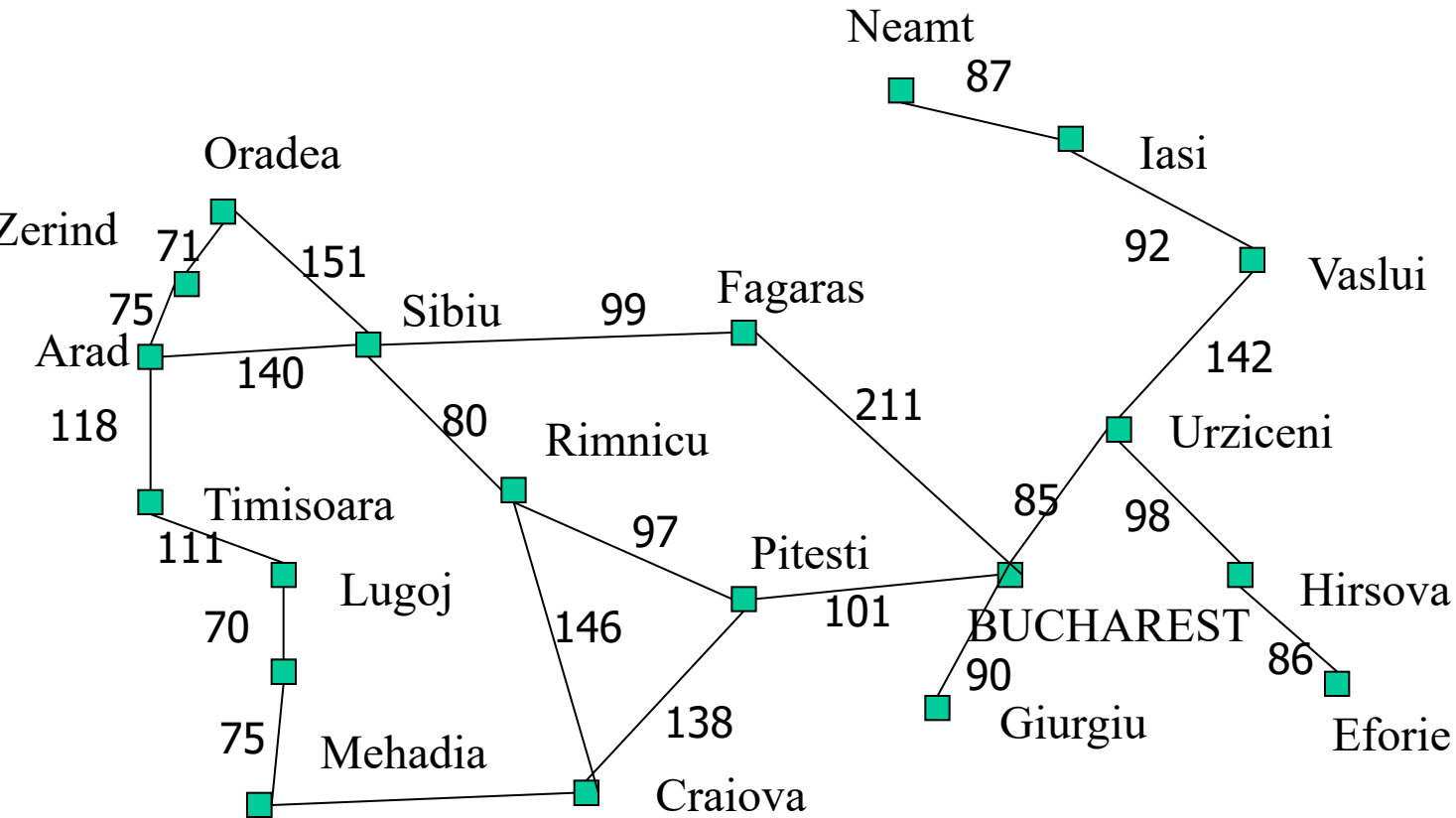


Distancia en  
línea recta a  
Bucarest:

Arad	366
Bucarest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

¿La distancia en línea recta es admisible?

# Búsqueda informada



La heurística es menor al valor real

Distancia en línea recta a Bucarest:	
Arad	366
Bucarest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

# Búsqueda informada

---

- $h(n)$ : número de placas en la posición incorrecta (incluyendo la placa vacía)
- Cada movimiento de la placa vacía cuesta 1

1	2	3
4	5	6
7		8

¿Es  $h(n)$  admisible?

# Búsqueda informada

---

- $h(n)$ : número de placas en la posición incorrecta (incluyendo la placa vacía)
- Cada movimiento de la placa vacía cuesta 1

1	2	3
4	5	6
7		8

$h=2$  pero el costo real es 1,  
por lo tanto  $h$  no es admisible

# Búsqueda informada

---

- $h(n)$ : número de placas en la posición incorrecta  
(sin incluir la placa vacía)

es admisible

# Búsqueda informada

---

- Una heurística es admisible si nunca asigna un valor mayor al real, es decir, se cumple que

$$h(n) \leq \text{costo\_real}(n)$$

# Búsqueda informada

---

- Una heurística es admisible si nunca asigna un valor mayor al real

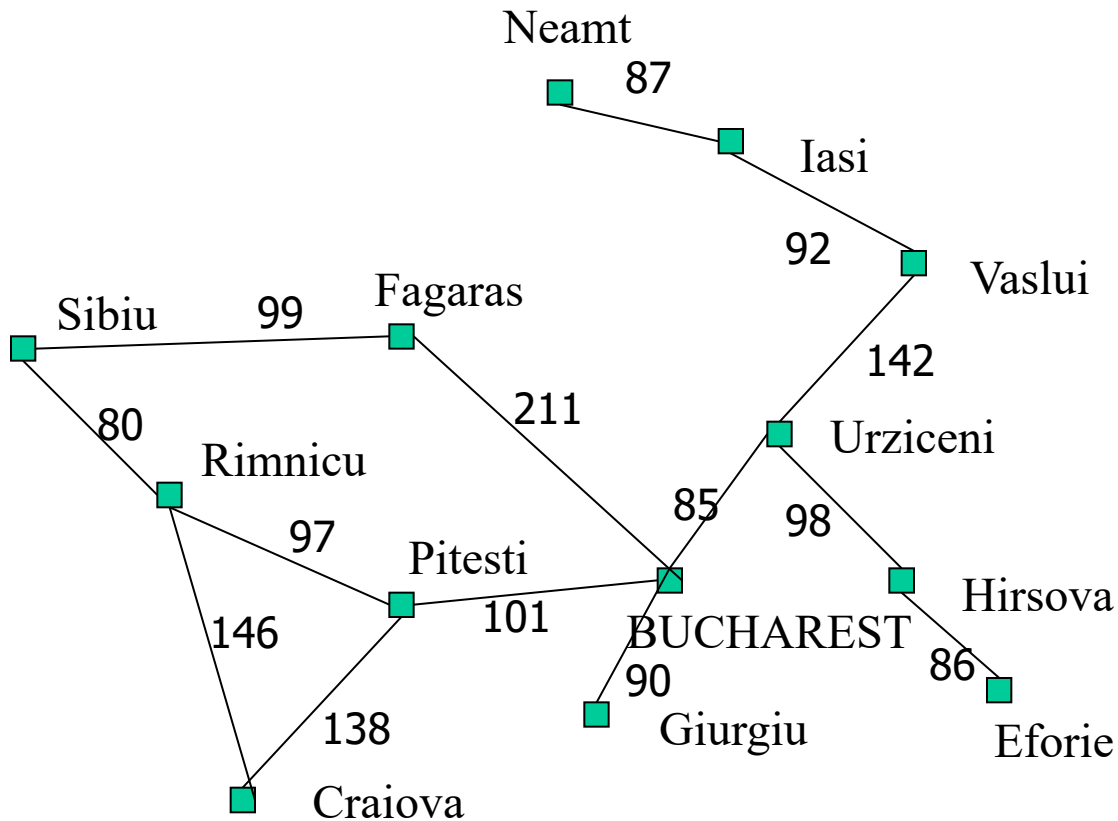
- Construya el árbol de búsqueda para ir de Sibiu a Bucarest utilizando  $h_1(n)$ =distancia en línea recta

- Construya el árbol de búsqueda para ir de Sibiu a Bucarest utilizando  $h_2(n)$ =distancia en línea recta\*2

Evite devolverse



# Búsqueda informada



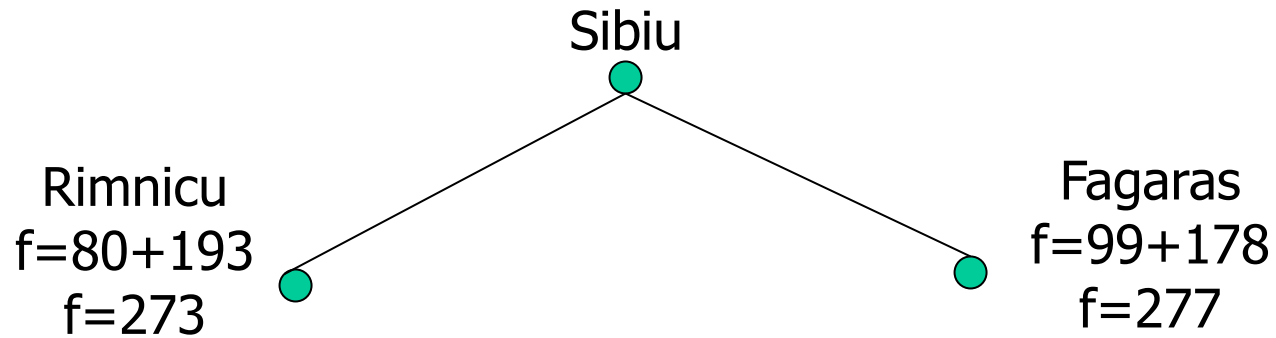
Distancia en  
línea recta a  
Bucarest:

Arad	366
Bucarest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

# Búsqueda informada

---

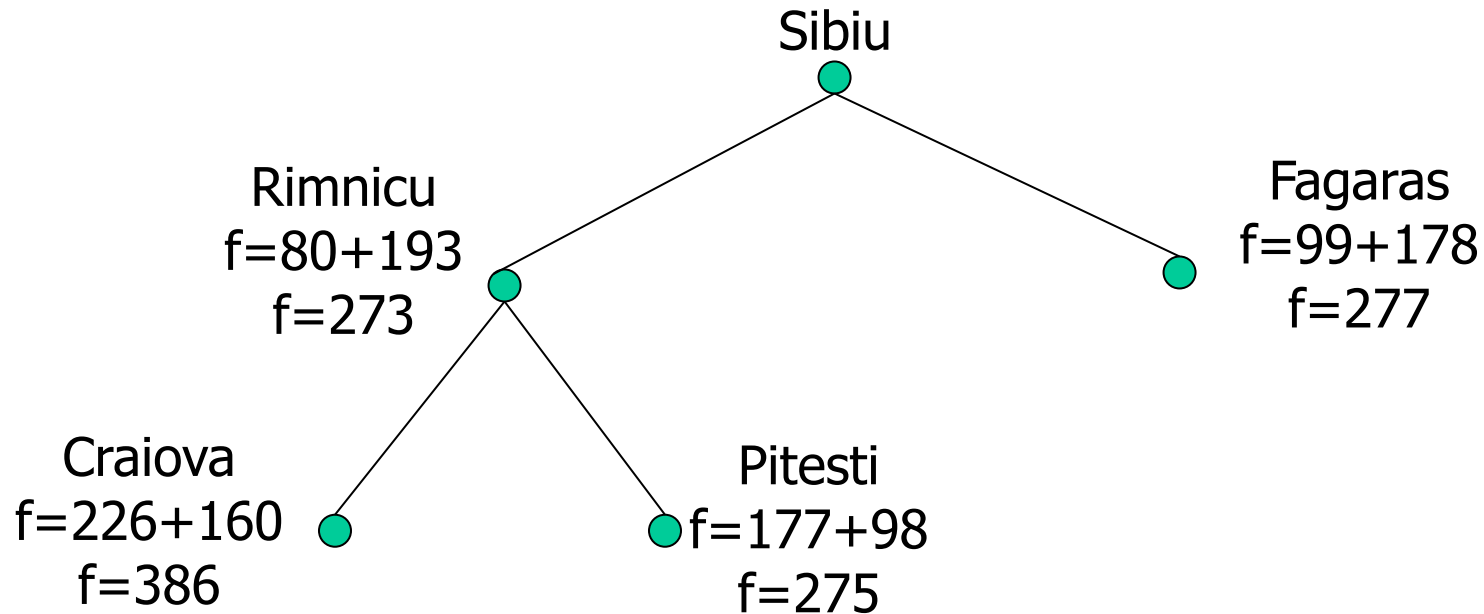
- $h_1(n)$ =distancia en línea recta



# Búsqueda informada

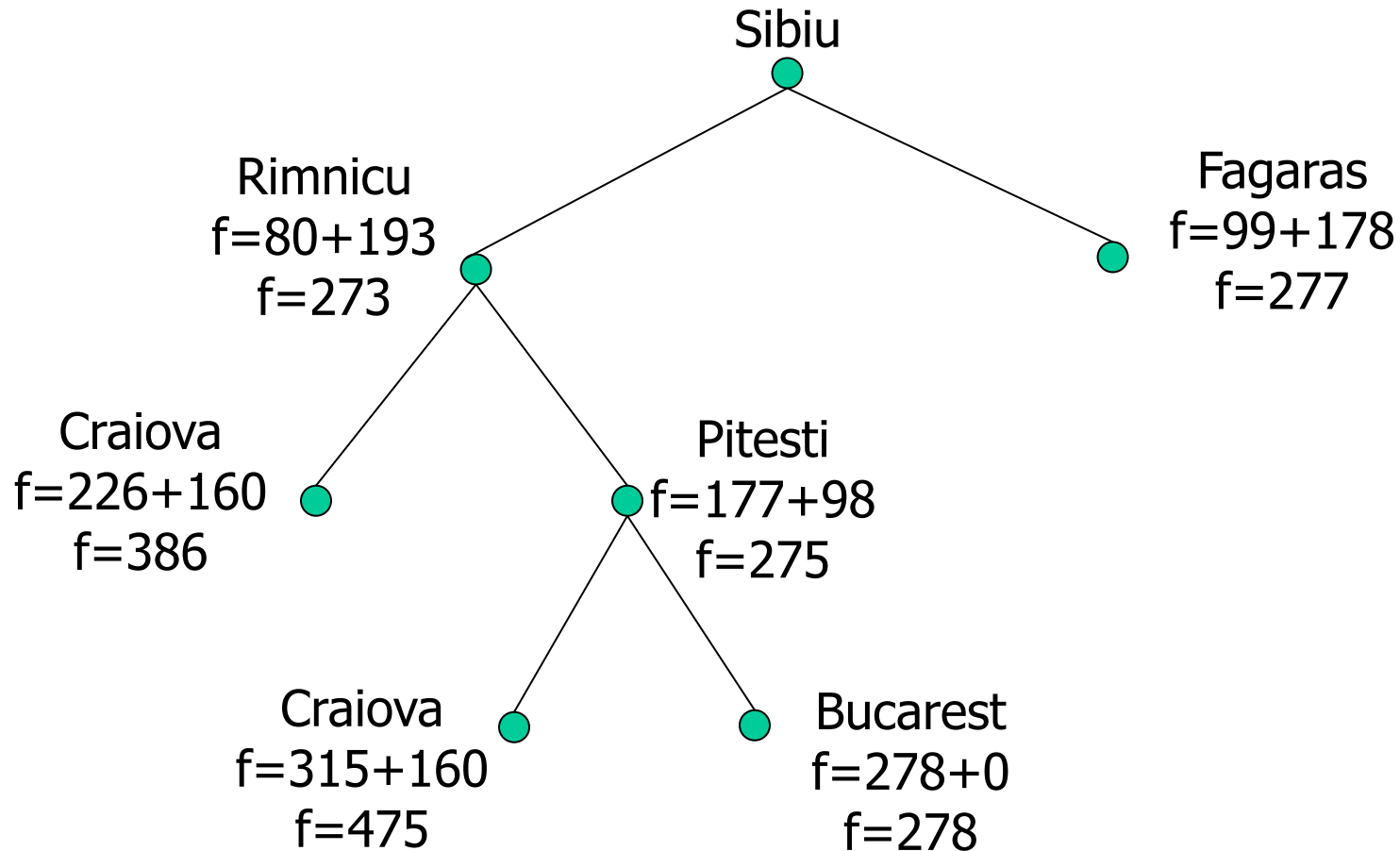
---

- $h_1(n)$ =distancia en línea recta



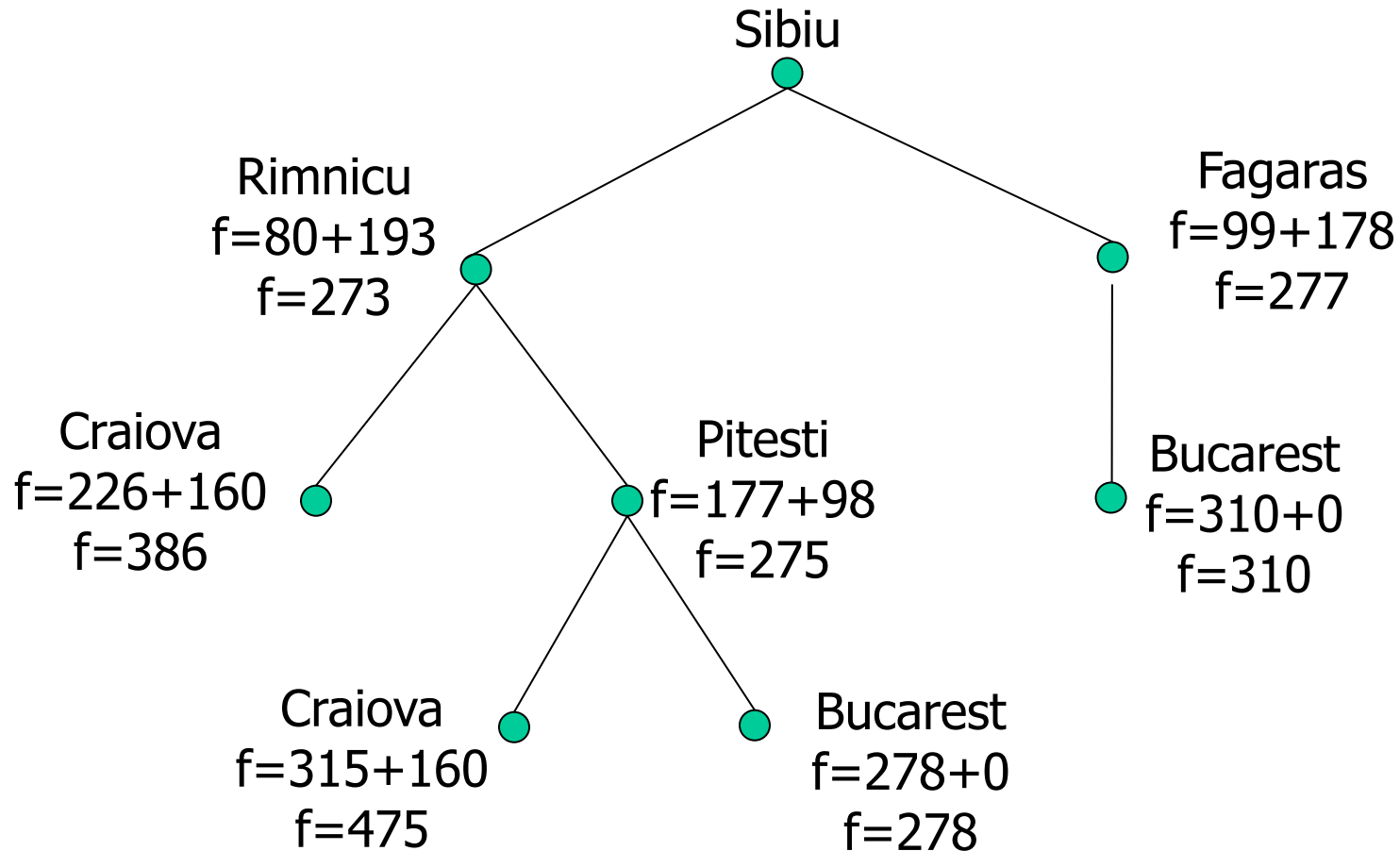
# Búsqueda informada

- $h_1(n)$ =distancia en línea recta



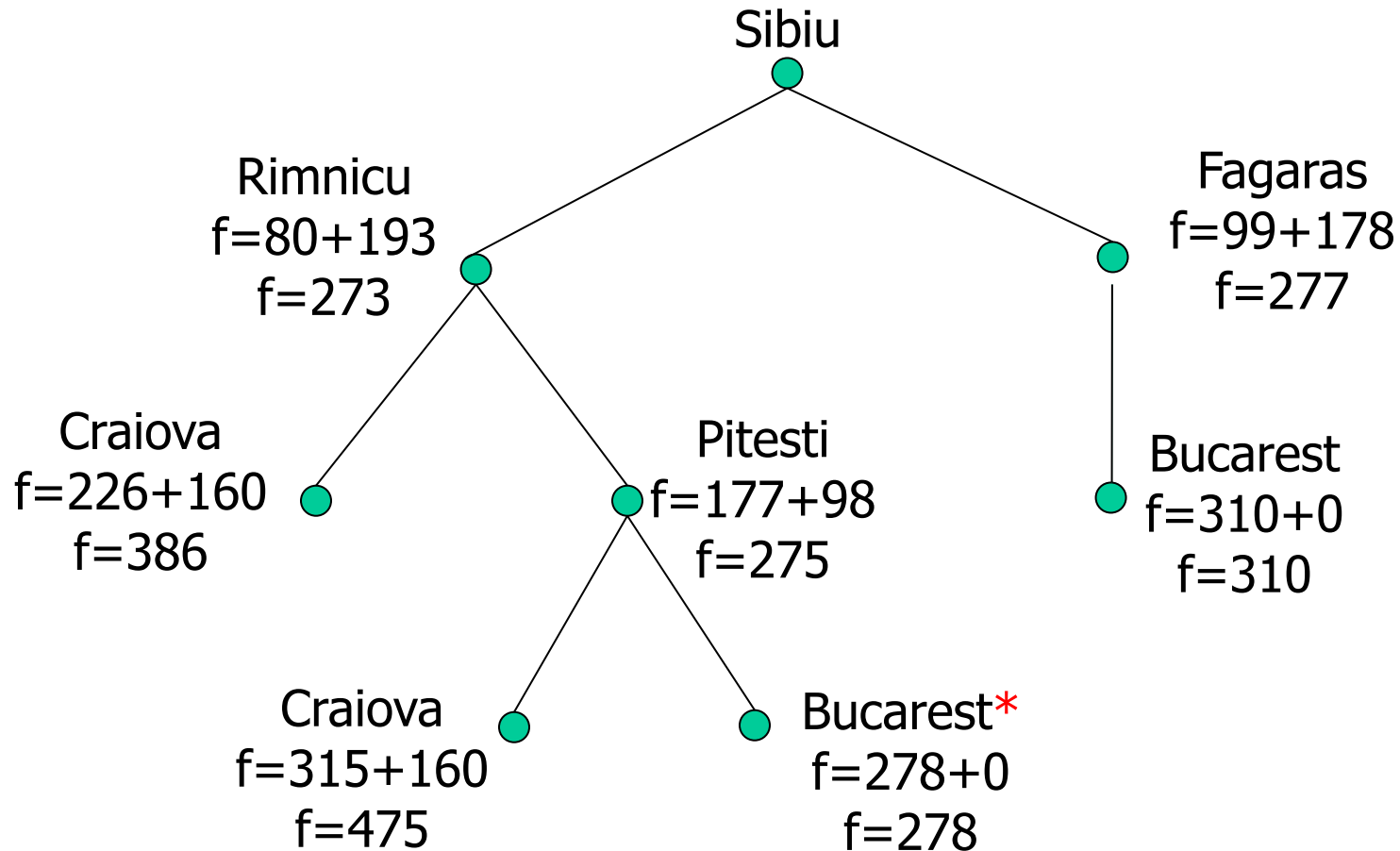
# Búsqueda informada

- $h_1(n)$ =distancia en línea recta



# Búsqueda informada

- $h_1(n)$ =distancia en línea recta



# Búsqueda informada

---

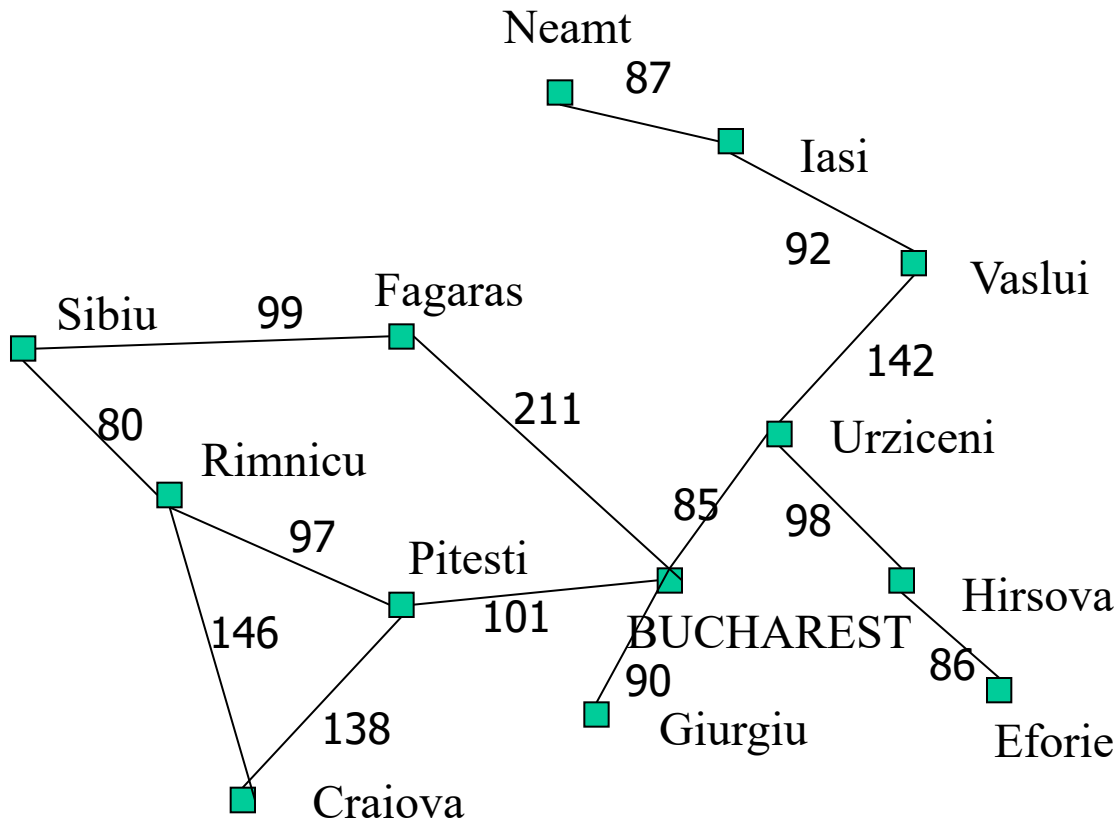
- Una heurística es admisible si nunca asigna un valor mayor al real

- Construya el árbol de búsqueda para ir de Sibiu a Bucarest utilizando  $h_1(n)$ =distancia en línea recta

- Construya el árbol de búsqueda para ir de Sibiu a Bucarest utilizando  $h_2(n)$ =distancia en línea recta\*2

Evite devolverse

# Búsqueda informada



Distancia en  
línea recta a  
Bucarest\*2:

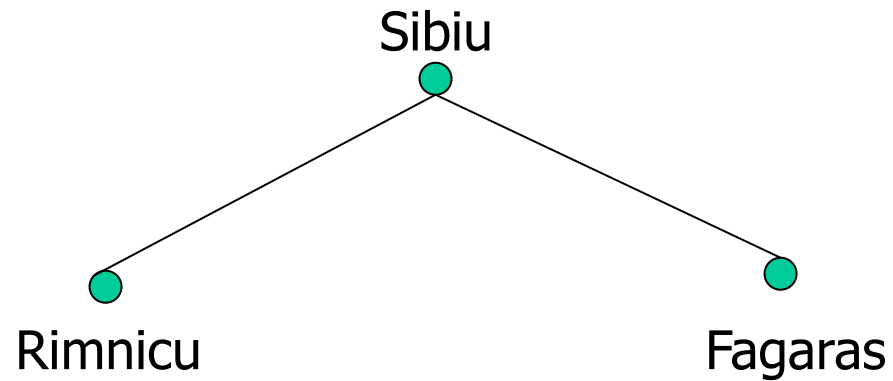
Arad	732
Bucarest	0
Craiova	320
Dobreta	484
Eforie	322
Fagaras	356
Giurgiu	154
Hirsova	302
Iasi	452
Lugoj	488
Mehadia	482
Neamt	468
Oradea	760
Pitesti	196
Rimnicu	386
Sibiu	506
Timisoara	658
Urziceni	160
Vaslui	398
Zerind	748



# Búsqueda informada

---

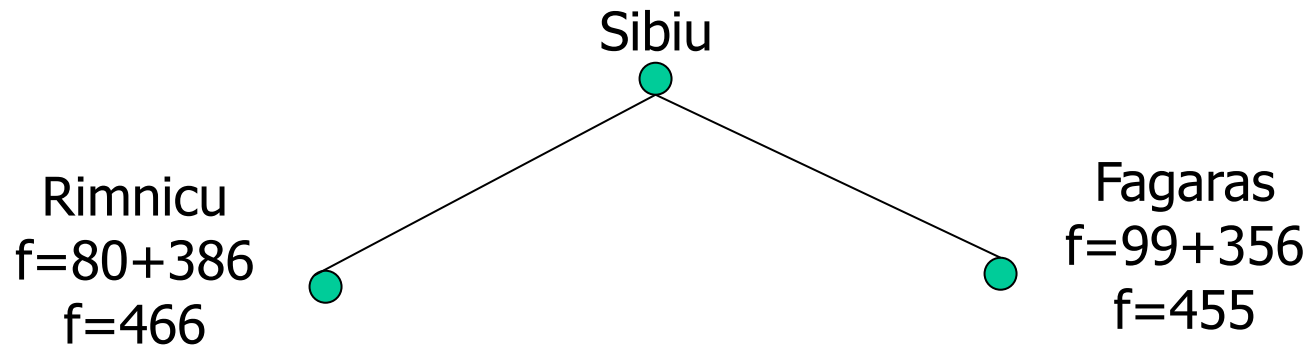
- $h_2(n)$ =distancia en línea recta\*2



# Búsqueda informada

---

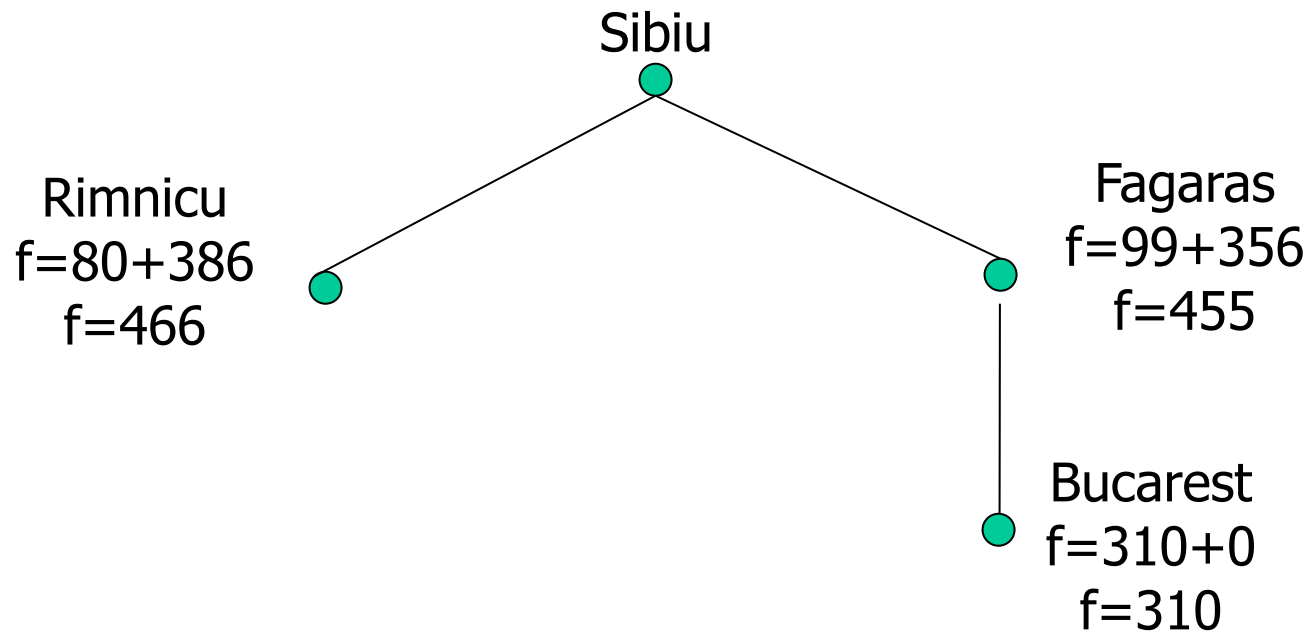
- $h_2(n) = \text{distancia en línea recta} * 2$



# Búsqueda informada

---

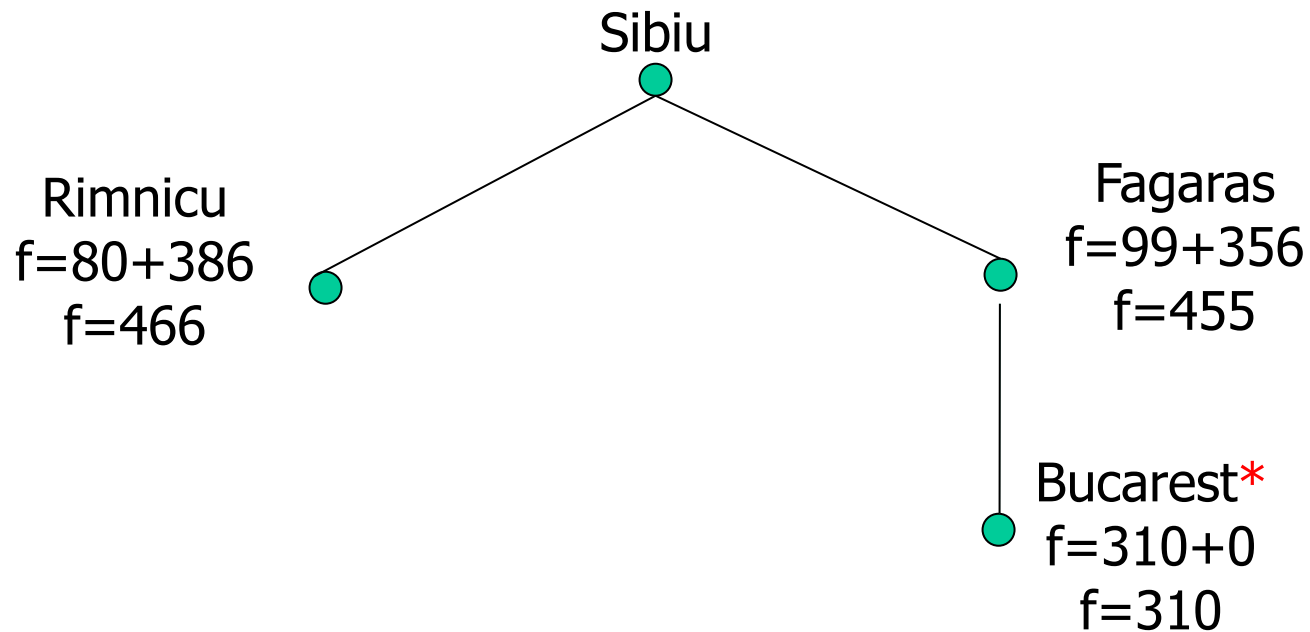
- $h_2(n)$ =distancia en línea recta\*2



# Búsqueda informada

---

- $h_2(n) = \text{distancia en línea recta} * 2$

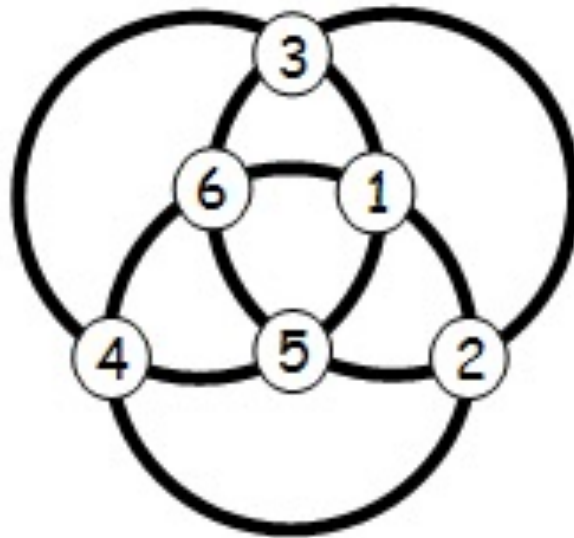


# Búsqueda informada

---

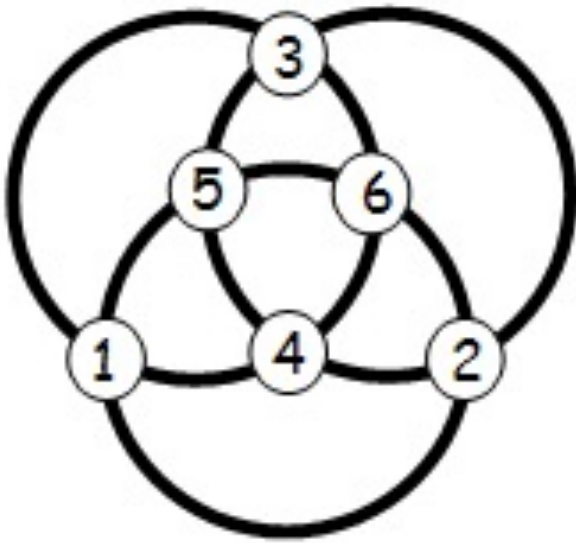
## Diseño de heurísticas

- **Los aros mágicos.** Utilizando el operador *intercambio(a,b)* que cambia de posición los números a y b, colocar los cuatro números sobre cada aro de tal forma que la suma sea 14. Cada intercambio cuesta 1

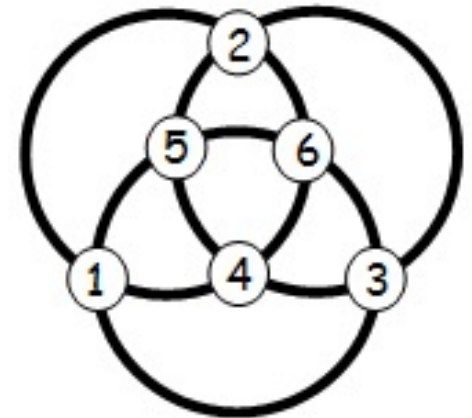
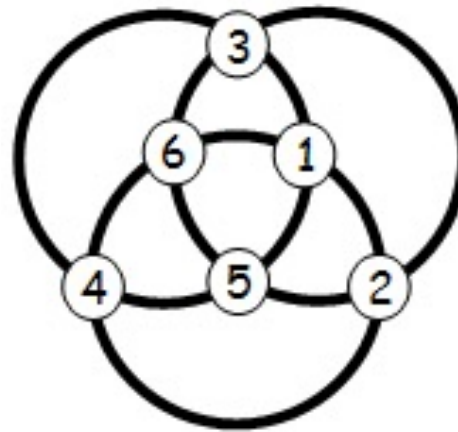


# Búsqueda informada

## Diseño de heurísticas



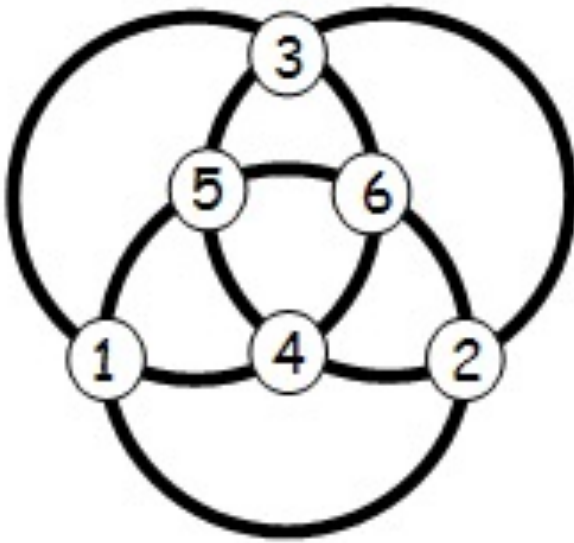
meta



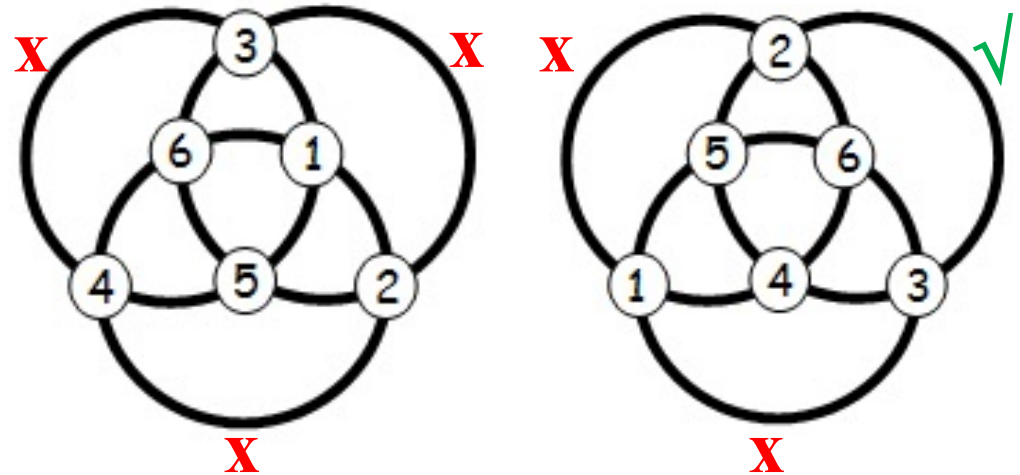
¿Cuál está más distante de la meta?  
¿Cómo lo sabe?

# Búsqueda informada

## Diseño de heurísticas



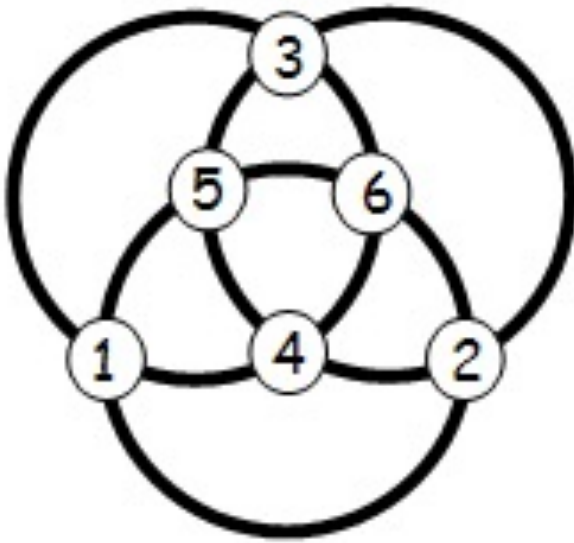
meta



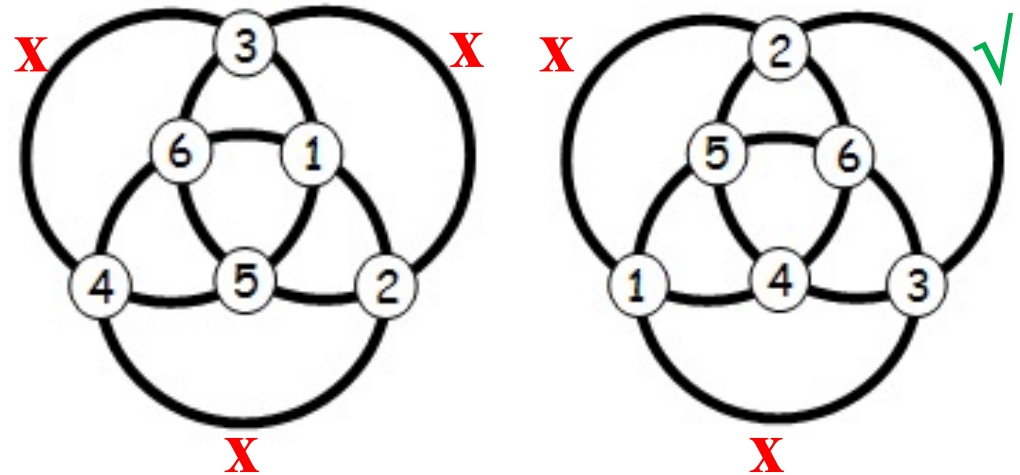
¿Cuál está más distante de la meta?  
¿Cómo lo sabe?

# Búsqueda informada

## Diseño de heurísticas



meta

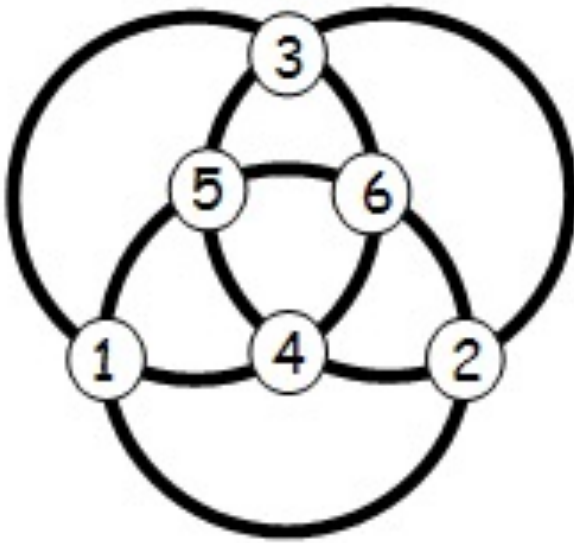


$h(n)$  = cantidad de aros que no suman 14  
es admisible?

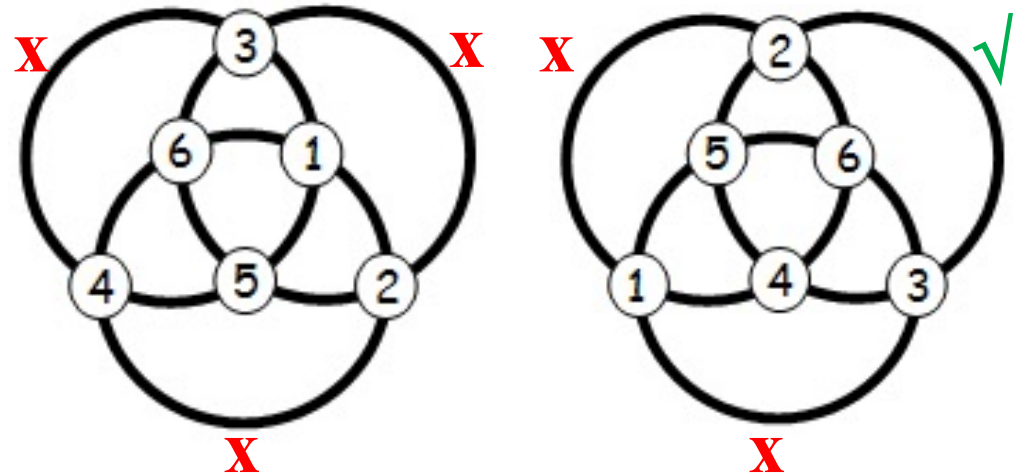


# Búsqueda informada

## Diseño de heurísticas



meta

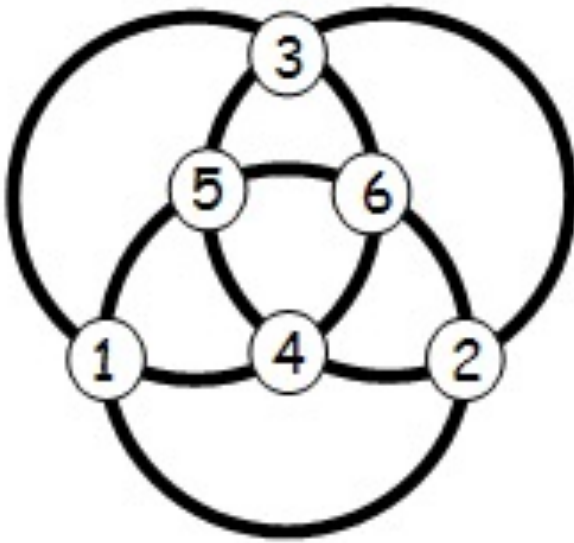


$h(n) = \text{cantidad de aros que no suman 14}$

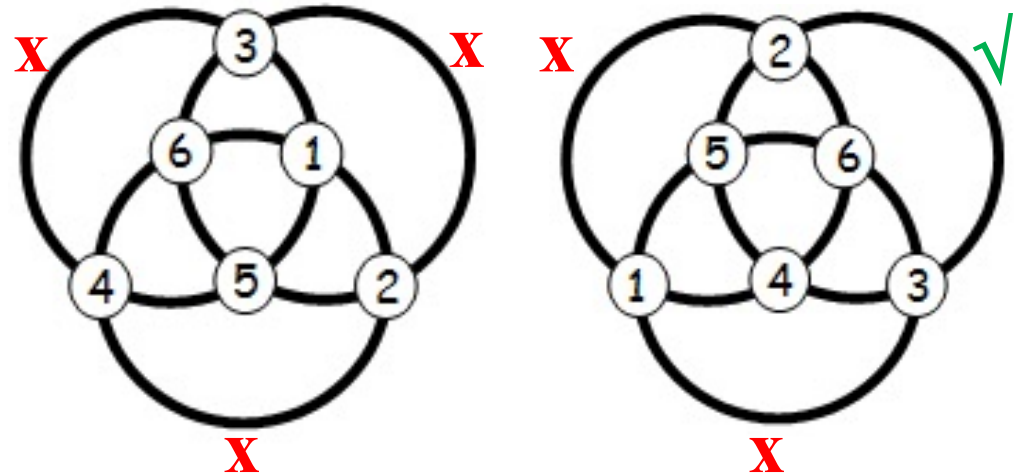
?

# Búsqueda informada

## Diseño de heurísticas



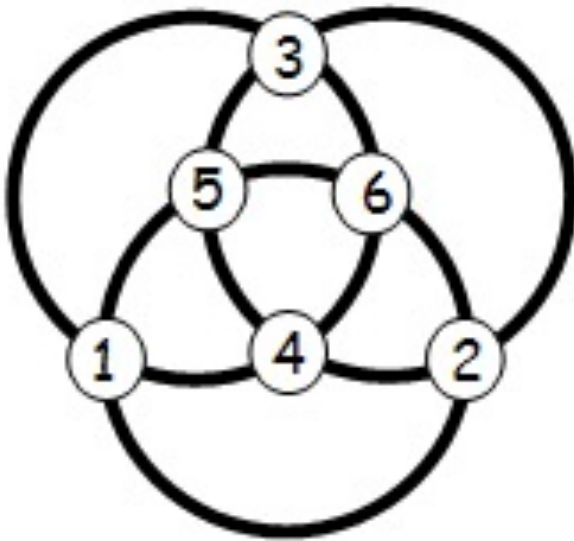
meta



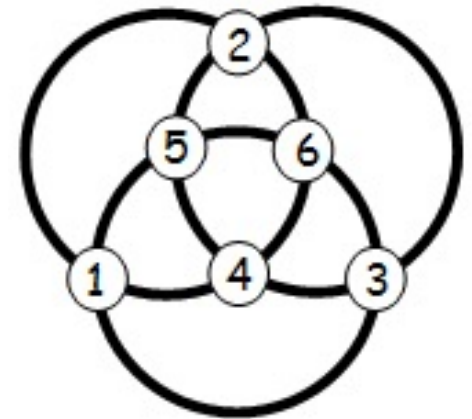
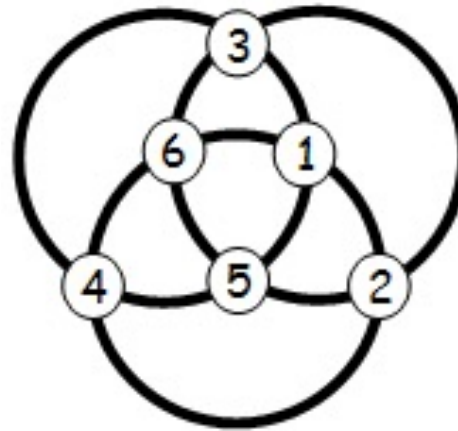
$$h(n) = \frac{\text{cantidad de aros que no suman 14}}{2}$$

# Búsqueda informada

## Diseño de heurísticas



meta



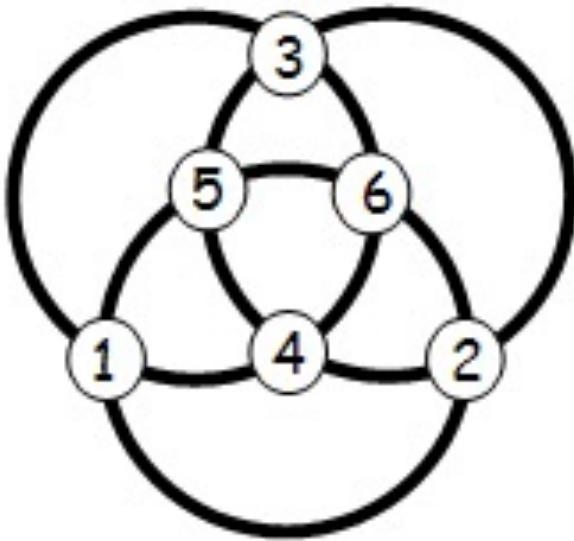
$h(n) = \frac{\text{cantidad de aros que no suman 14}}{2}$

2

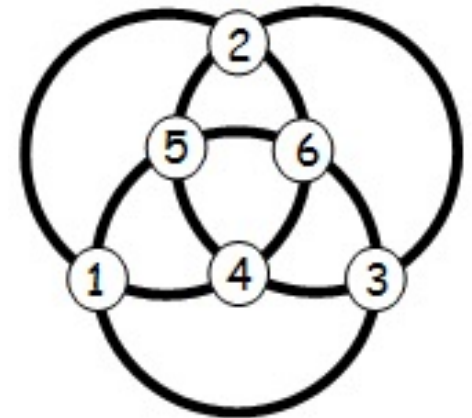
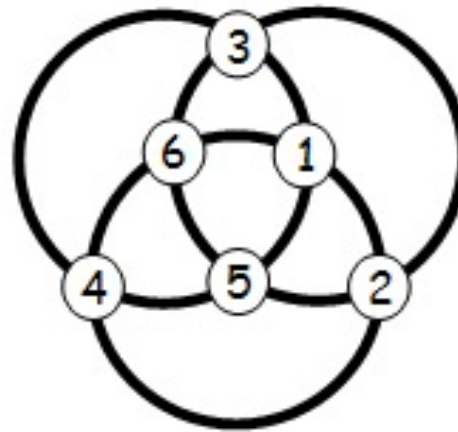
Se divide entre el mejor  
avance posible en un solo paso

# Búsqueda informada

## Diseño de heurísticas



meta

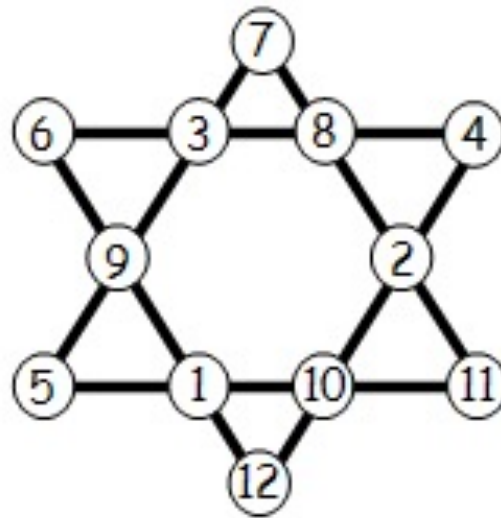


$$h(n) = \frac{\text{cantidad de aros que no suman 14}}{100}$$

# Búsqueda informada

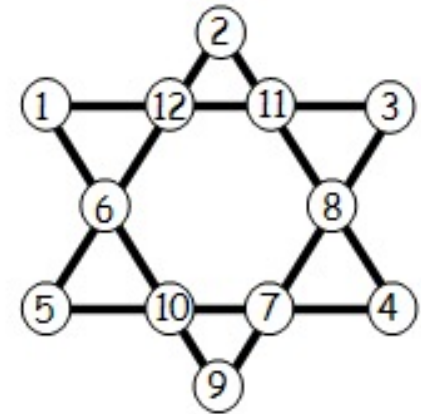
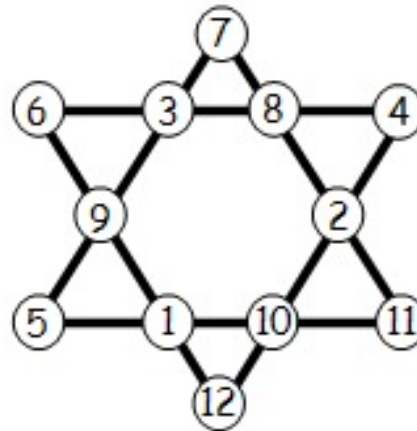
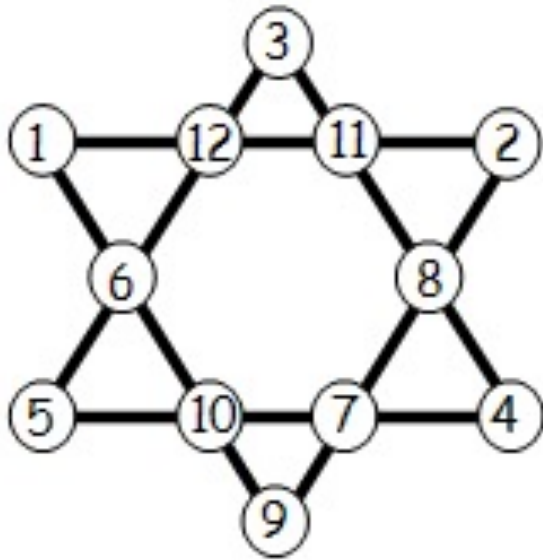
## Diseño de heurísticas

- **El número 26.** Utilizando el operador *intercambio(a,b)* que cambia de posición los números a y b, colocar los números de tal forma que sobre cada lado la suma sea 26. Cada intercambio cuesta 1.



# Búsqueda informada

## Diseño de heurísticas

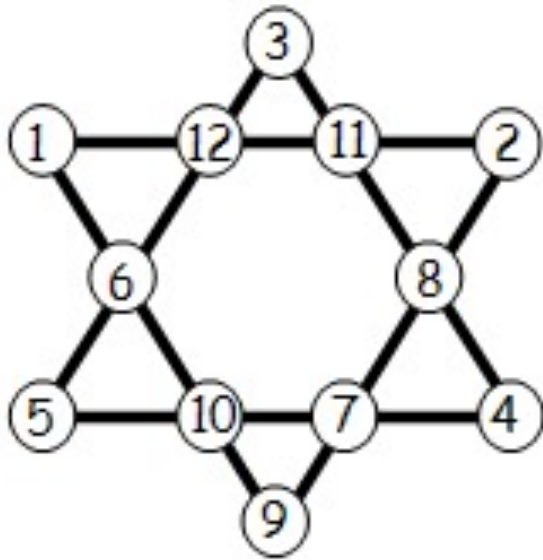


¿Cuál está más distante de la meta?

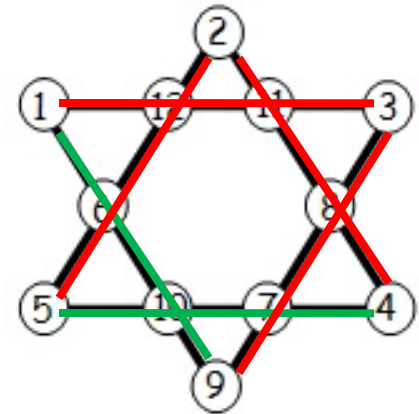
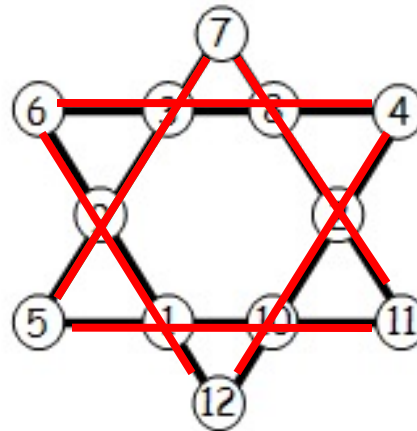
¿Cómo lo sabe?

# Búsqueda informada

## Diseño de heurísticas



meta

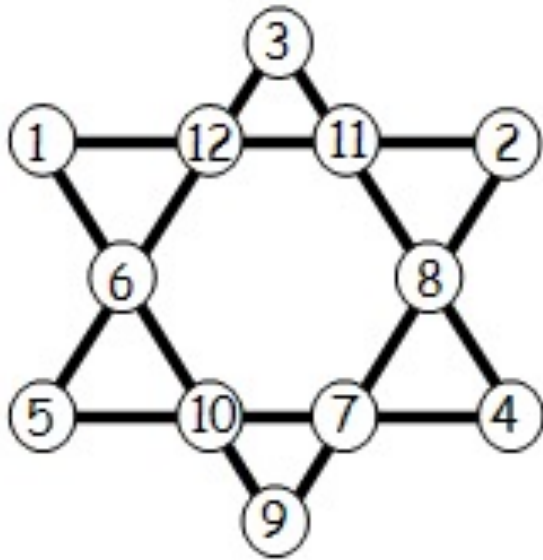


¿Cuál está más distante de la meta?

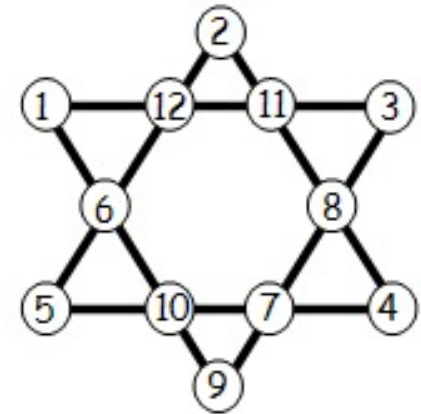
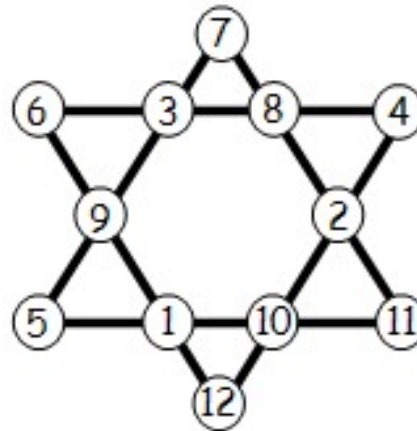
¿Cómo lo sabe?

# Búsqueda informada

## Diseño de heurísticas



meta



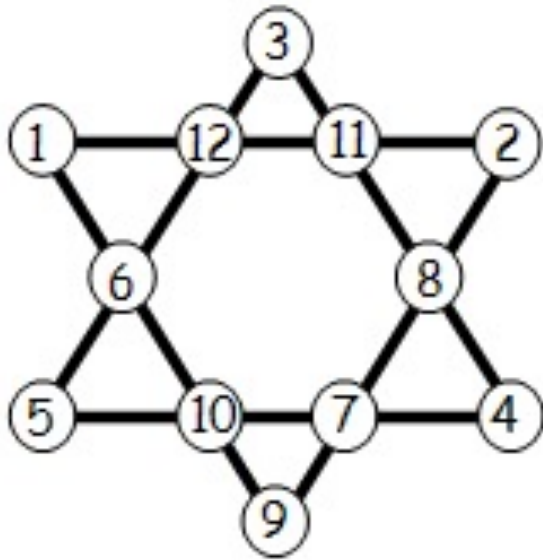
$h(n) = \text{cantidad de lados que no suman 26}$

?

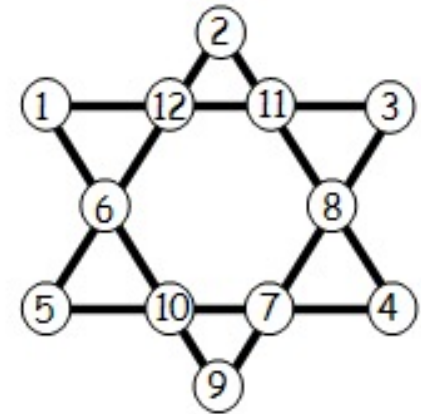
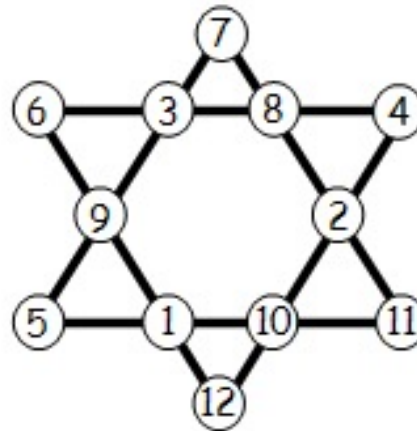


# Búsqueda informada

## Diseño de heurísticas



meta

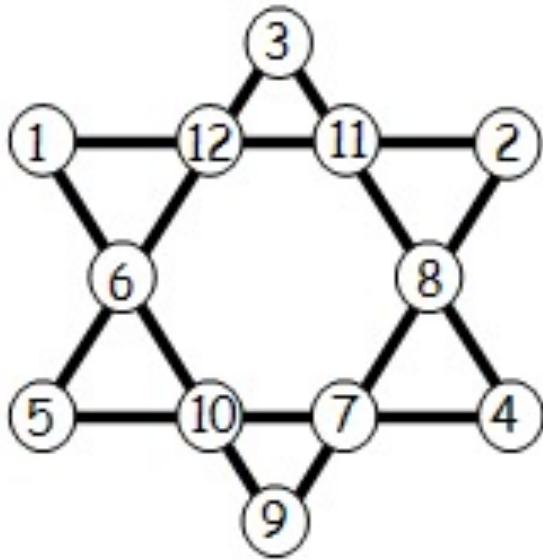


$$h(n) = \frac{\text{cantidad de lados que no suman 26}}{4}$$

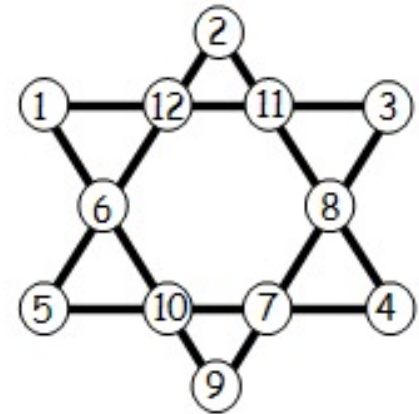
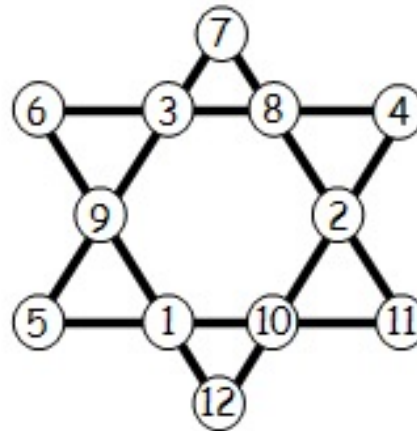
4

# Búsqueda informada

# Diseño de heurísticas



# meta



$$h(n) = \frac{\text{cantidad de lados que no suman 26}}{100}$$

# Búsqueda informada

---

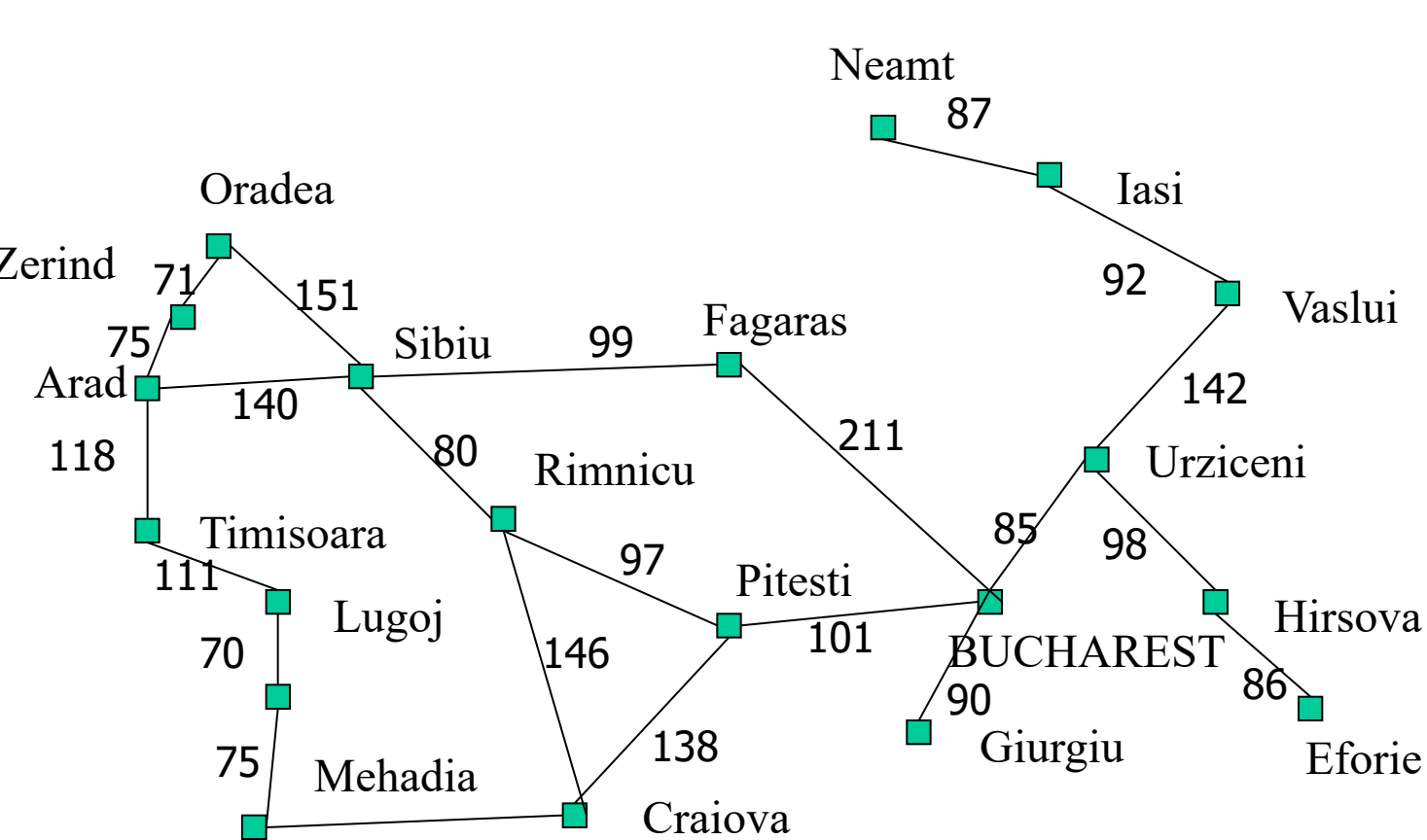
- **Complejidad:** si
- **Complejidad temporal:** número de nodos con una  $f(n)$  más pequeña que el costo óptimo
- **Complejidad espacial:** número de nodos con una  $f(n)$  más pequeña que el costo óptimo
- **Solución óptima:** si

# Búsqueda informada

---

Heurísticas y desempeño

# Búsqueda informada



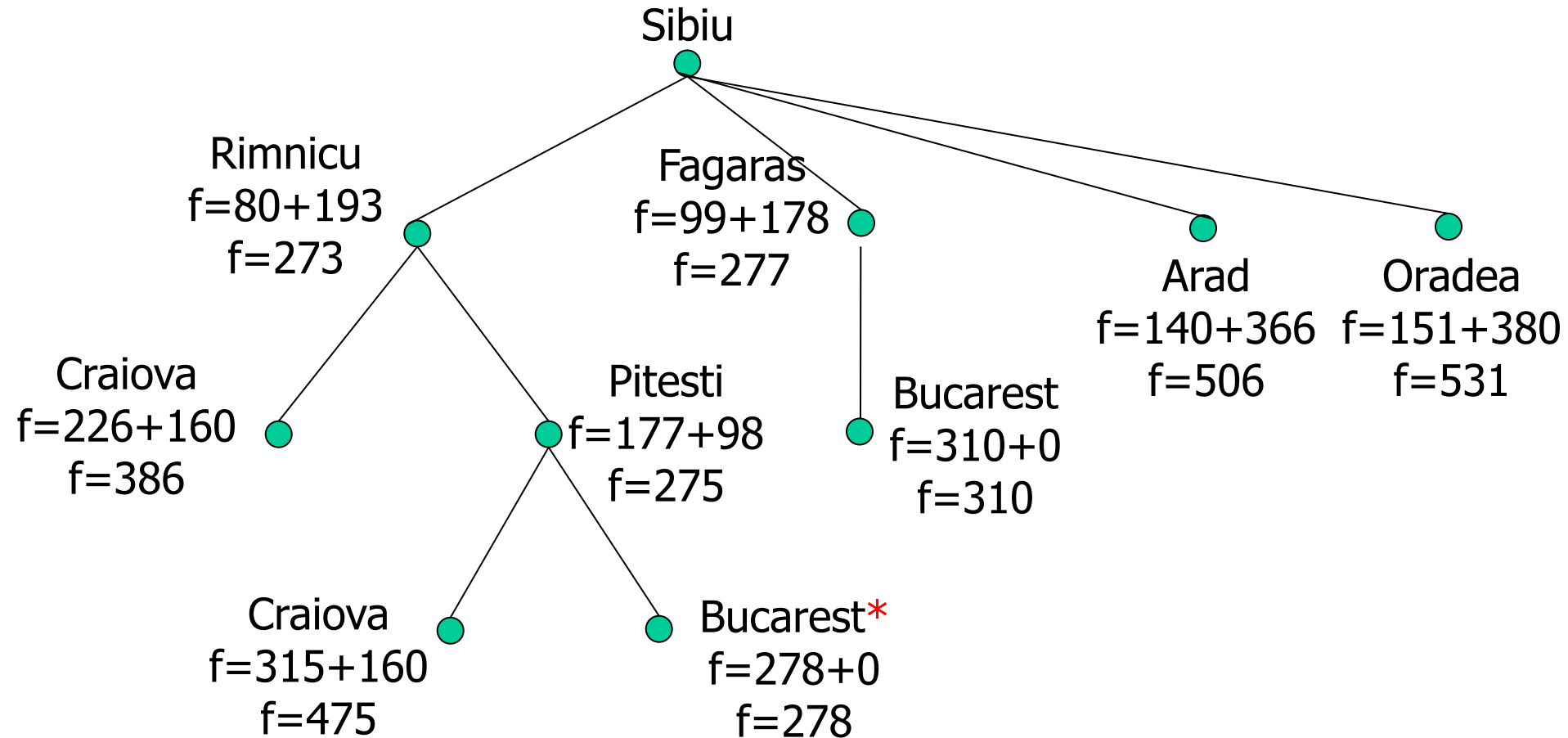
Distancia en  
línea recta a  
Bucarest:

Arad	366
Bucarest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

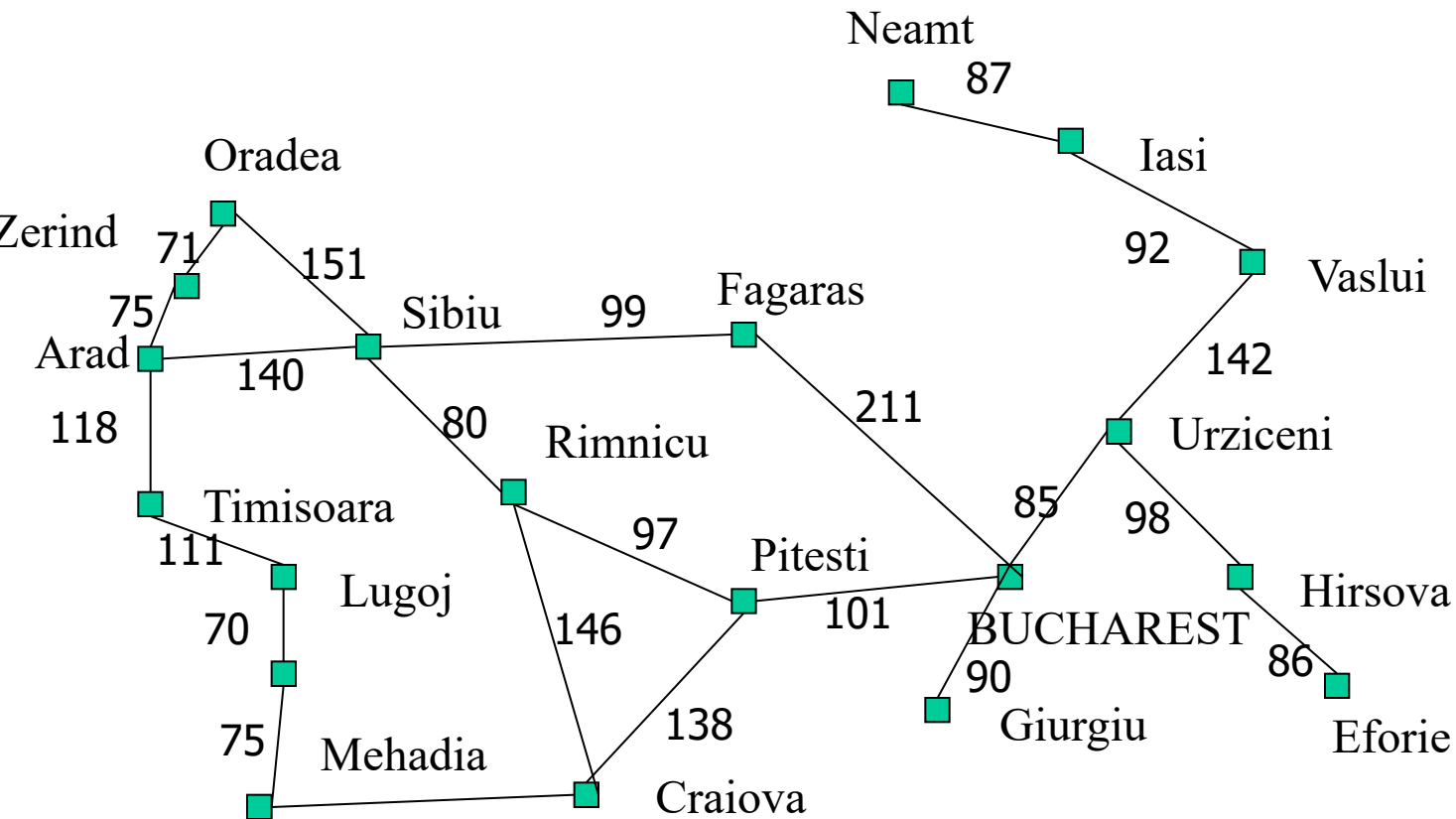
Problema: ir de Sibiu a Bucarest

# Búsqueda informada

---



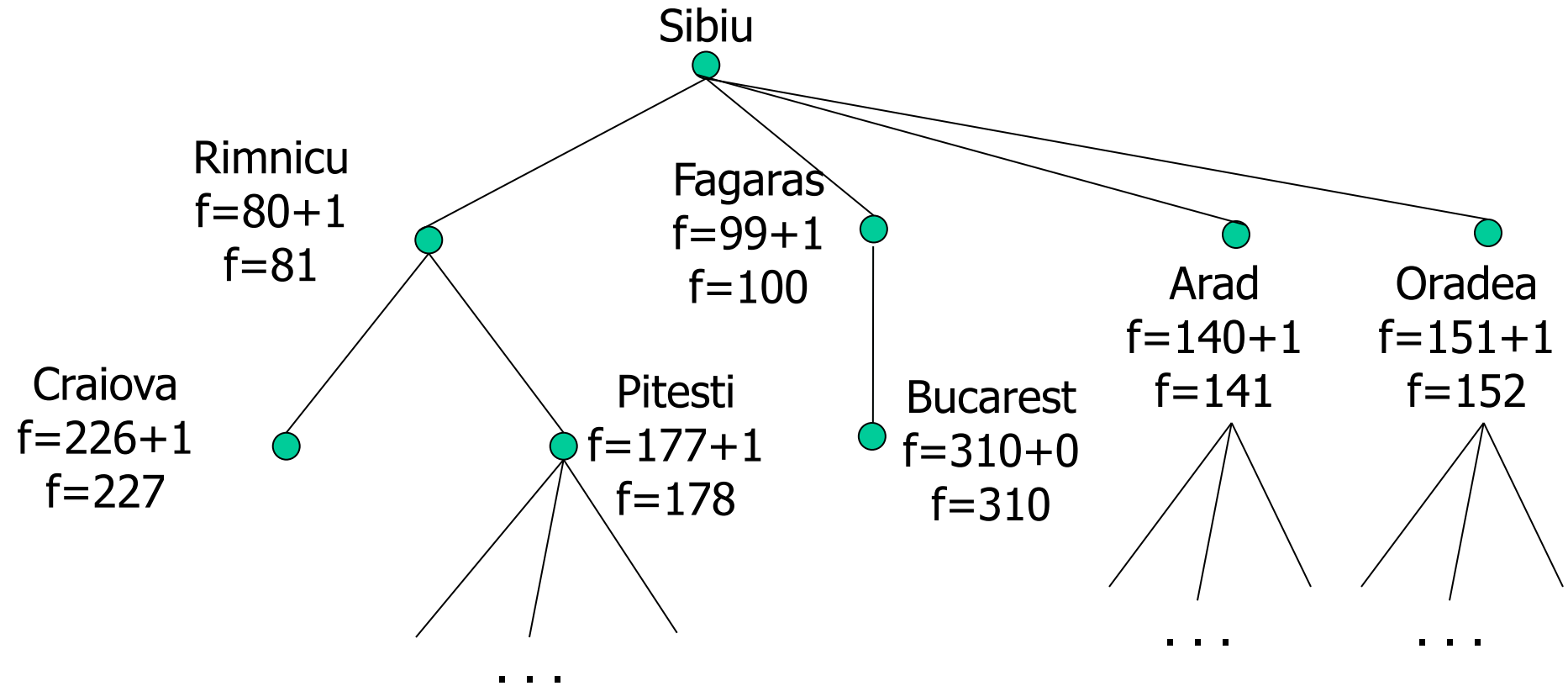
# Búsqueda informada



## Distancia mínima a Bucarest:

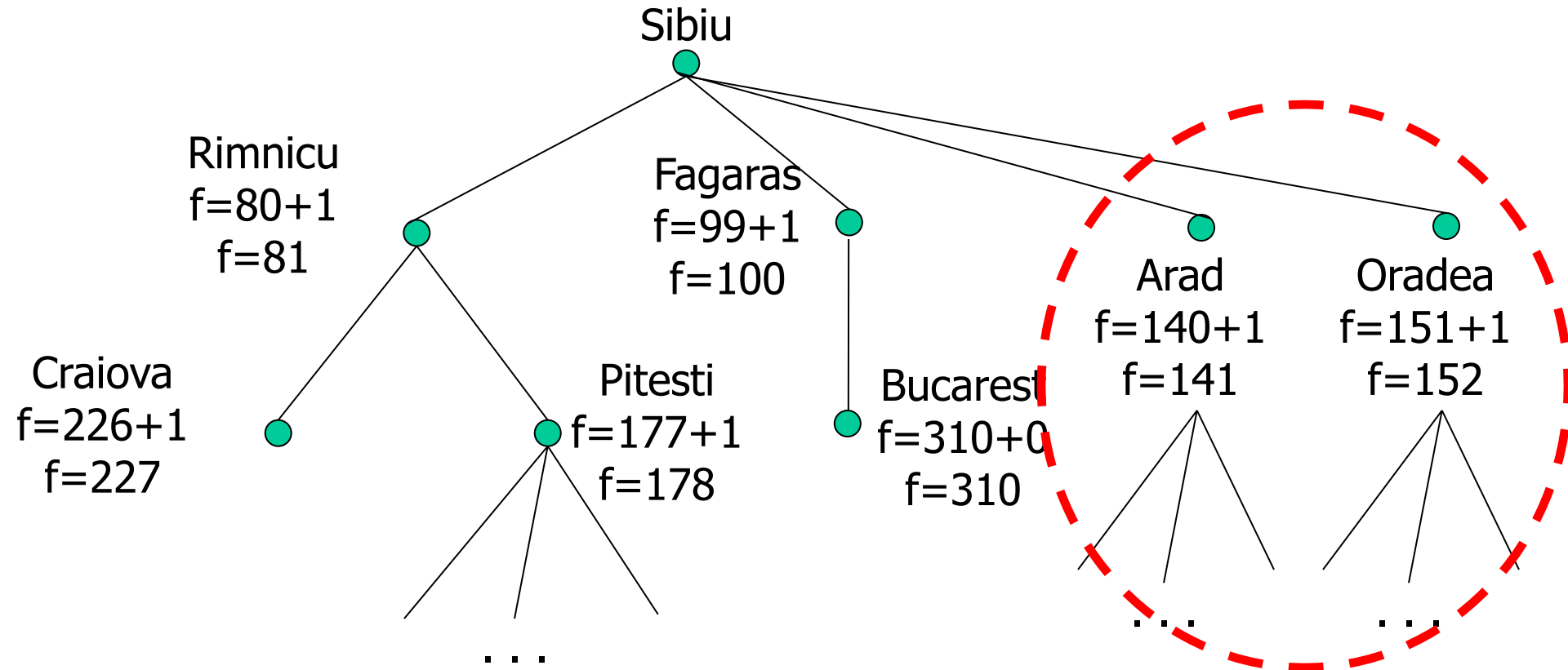
Arad	1
Bucarest	0
Craiova	1
Dobreta	1
Eforie	1
Fagaras	1
Giurgiu	1
Hirsova	1
Iasi	1
Lugoj	1
Mehadia	1
Neamt	1
Oradea	1
Pitesti	1
Rimnicu	1
Sibiu	1
Timisoara	1
Urziceni	1
Vaslui	1
Zerind	1

# Búsqueda informada

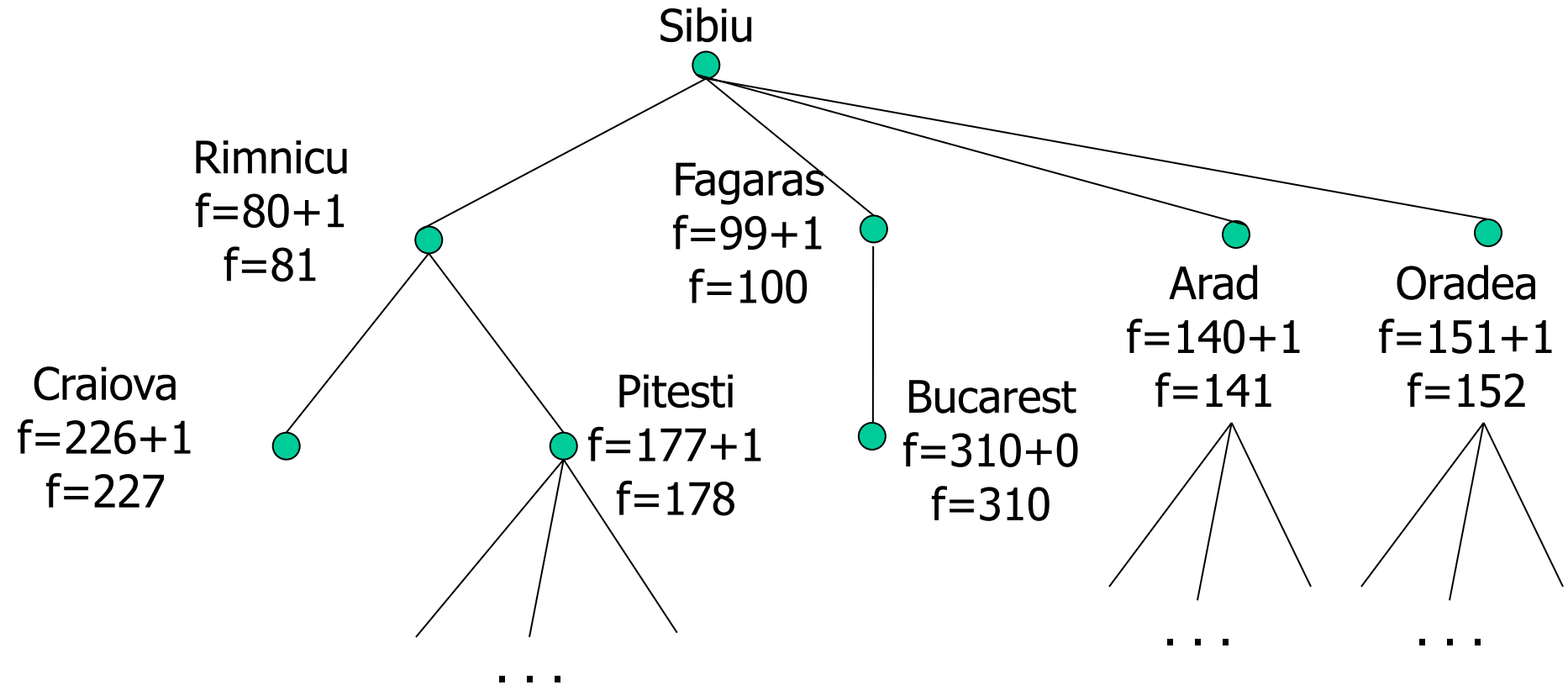




# Búsqueda informada



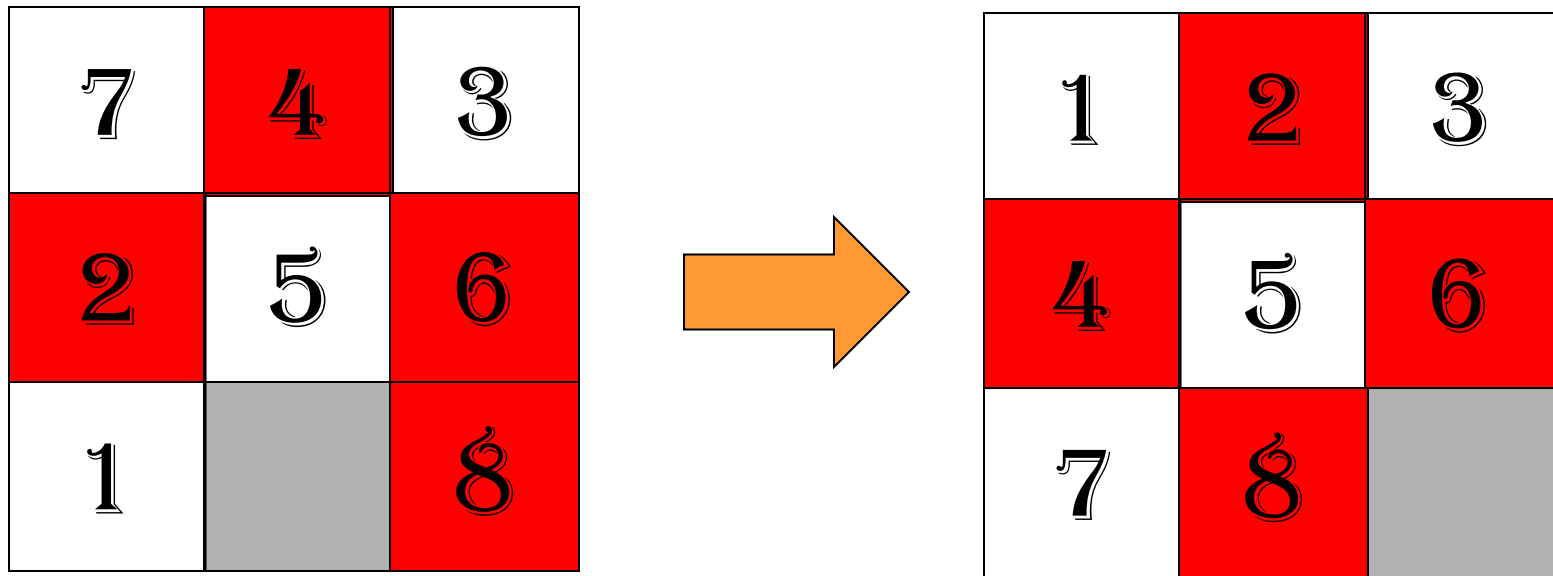
# Búsqueda informada



Si los valores de la heurística son muy pequeños,  $A^*$  se vuelve como Costo uniforme

# Búsqueda informada

## 8-puzzle



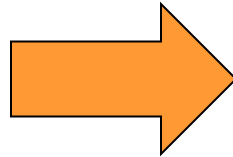
$h_1(n)$ : cantidad de piezas ubicadas en el lugar incorrecto (sin incluir la placa vacía)

# Búsqueda informada

## 8-puzzle

7	4	3
2	5	6
1		8

$h_1(n)$ : 5

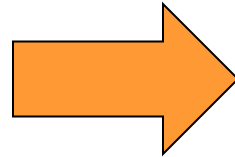


1	2	3
4	5	6
7	8	

# Búsqueda informada

## 8-puzzle

7	4	3
2	5	6
1		8



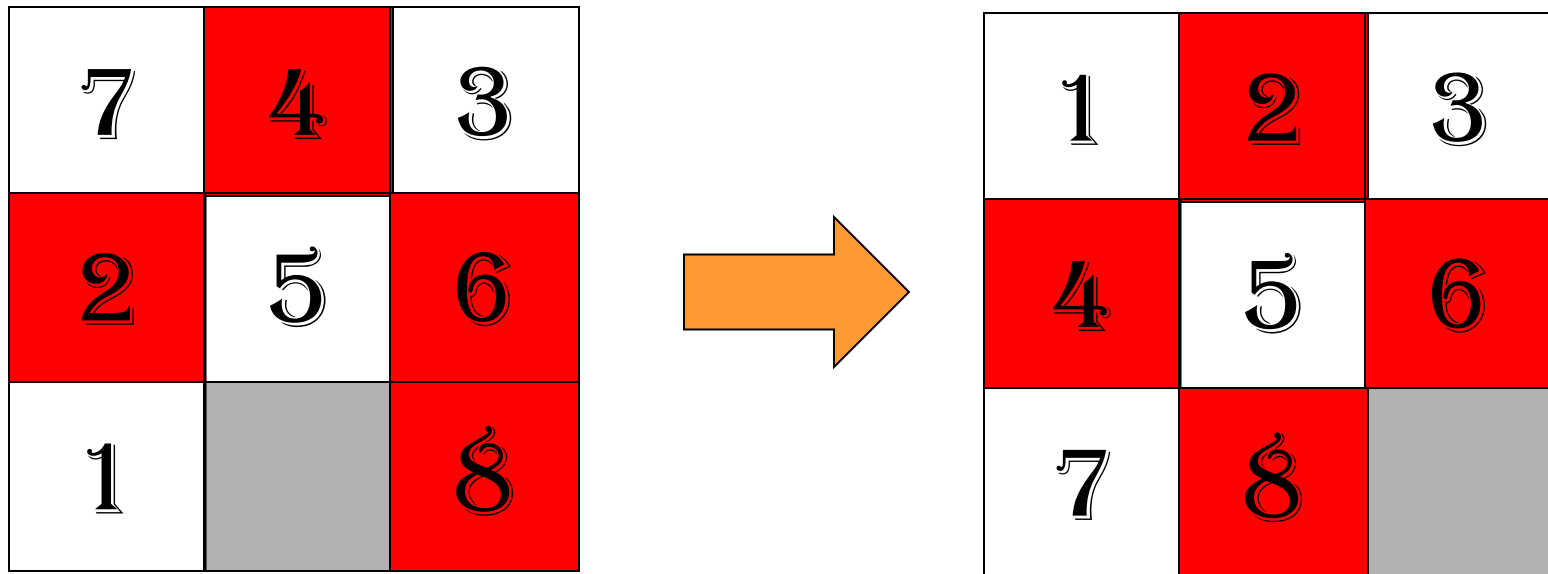
1	2	3
4	5	6
7	8	

$h_1(n): 5$

Realmente son muchos más que 5 movimientos  
El valor estimado está muy por debajo del valor real

# Búsqueda informada

## 8-puzzle

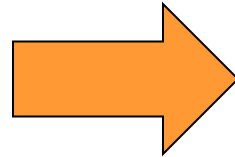


$h_2(n)$ : suma de las distancias de Manhattan de cada pieza a su posición correcta (sin incluir la placa vacía)

# Búsqueda informada

## 8-puzzle

7	4	3
2	5	6
1		8



1	2	3
4	5	6
7	8	

$$h_2(n): 2+2+0+2+0+0+2+1=9$$

# Búsqueda informada

---

## 8-puzzle

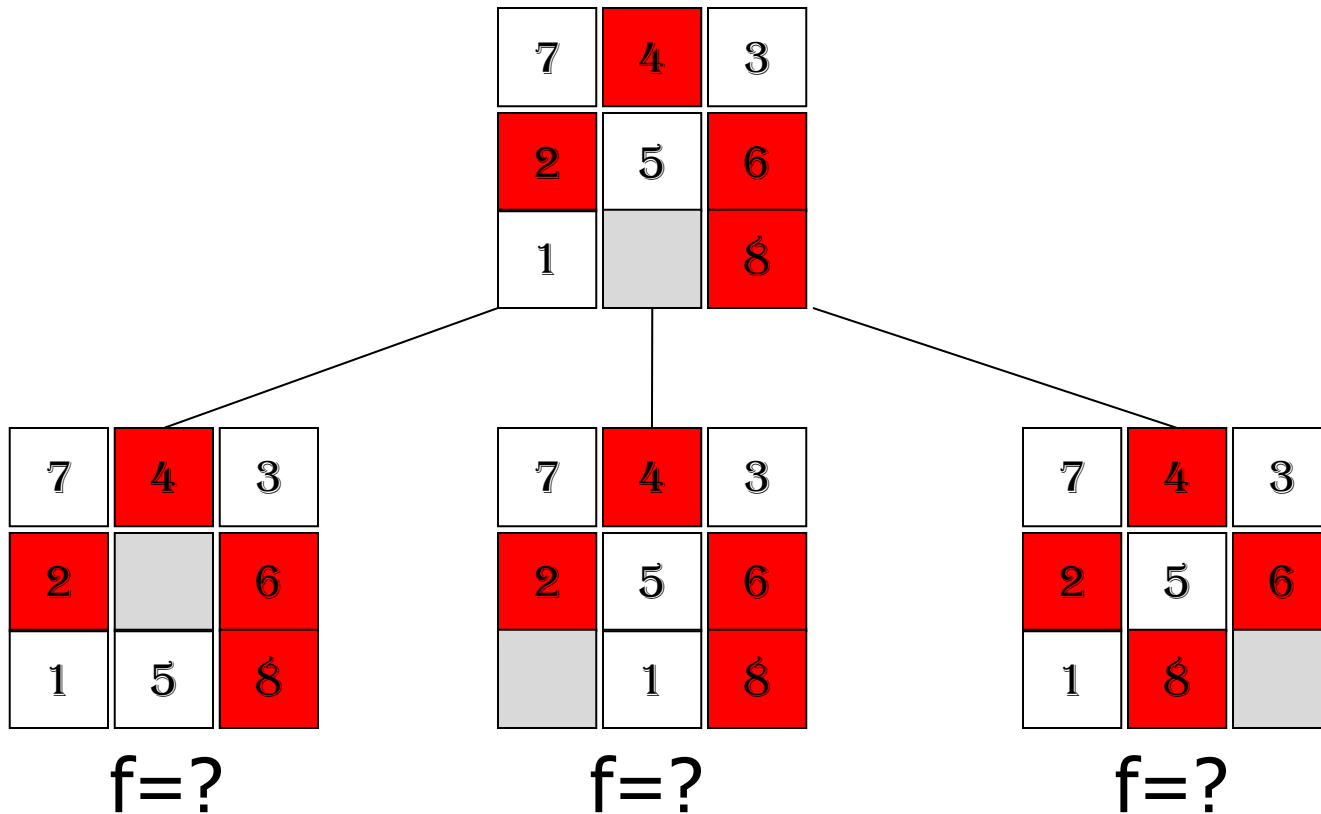
$h_1(n)$ : cantidad de piezas ubicadas en el lugar incorrecto

$h_2(n)$ : suma de la distancias horizontales y verticales que separan a las placas de las posiciones meta (excluyendo la vacía). **Distancia de Manhattan**

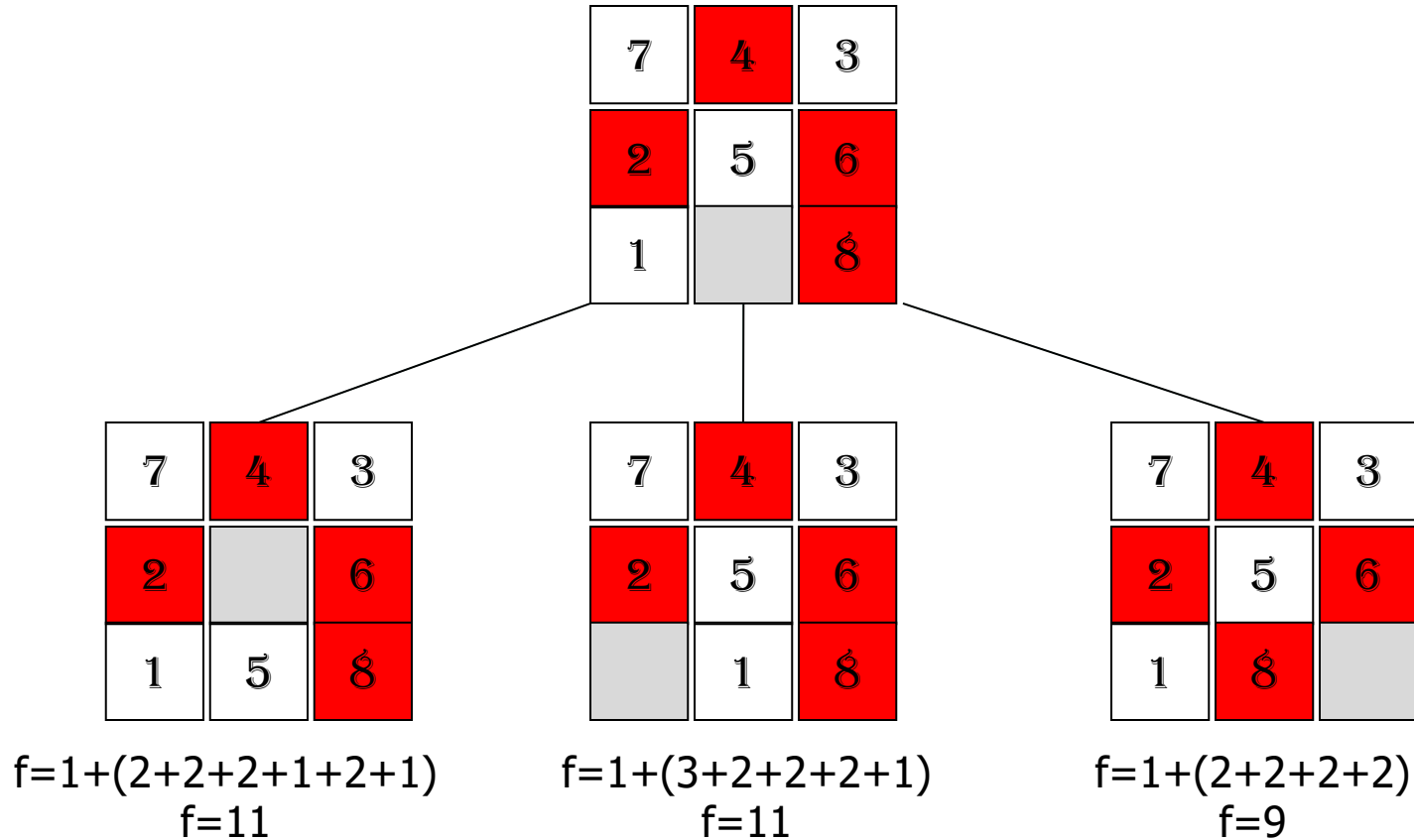


# Búsqueda informada

---



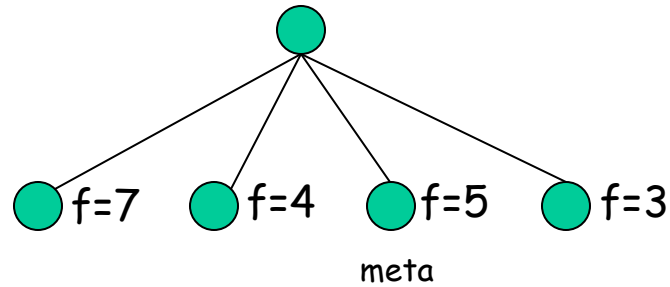
# Búsqueda informada



# Búsqueda informada

---

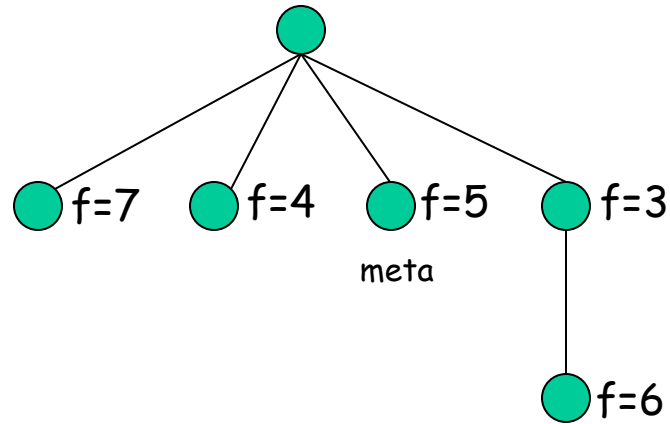
## Heurísticas y desempeño



# Búsqueda informada

---

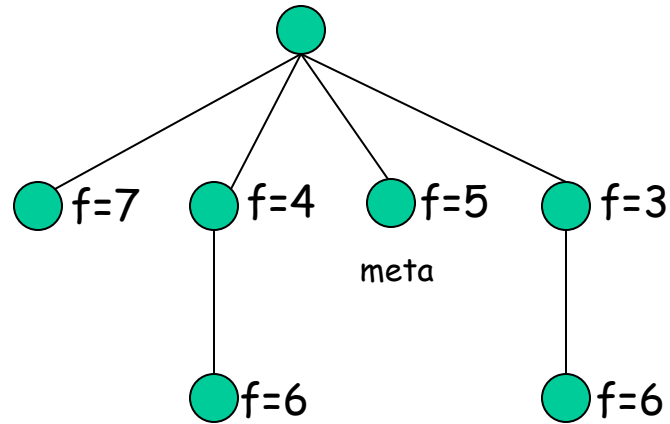
## Heurísticas y desempeño



# Búsqueda informada

---

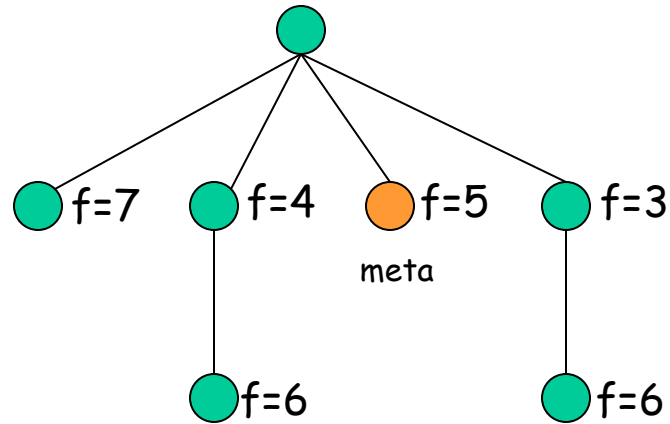
## Heurísticas y desempeño



# Búsqueda informada

---

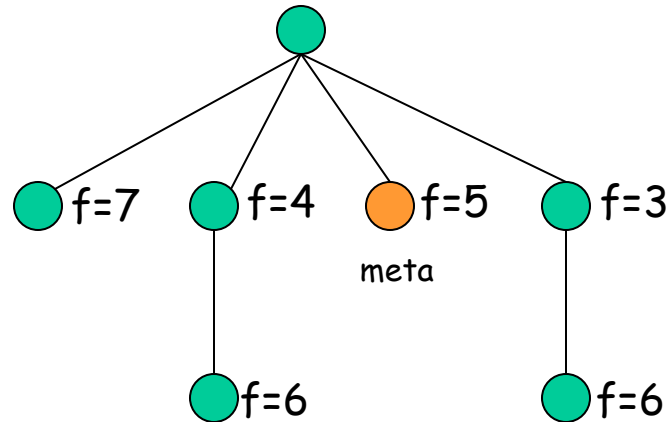
## Heurísticas y desempeño



# Búsqueda informada

---

## Heurísticas y desempeño



¿Cuál es el factor de  
ramificación del árbol?

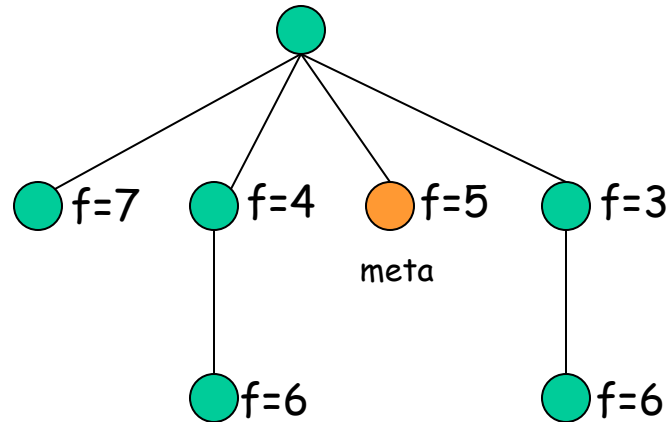
# Búsqueda informada

## Heurísticas y desempeño

$$N = 7$$

$$d = 2$$

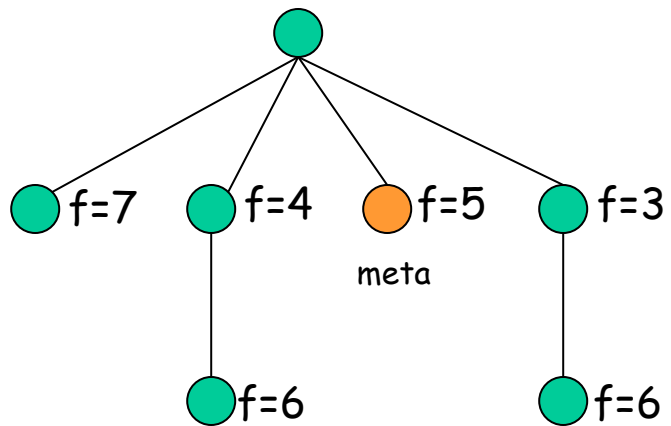
$$1 + b^* + (b^*)^2 = 7$$



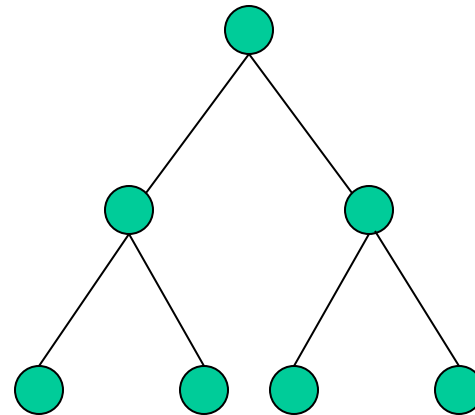


# Búsqueda informada

## Heurísticas y desempeño



*equivale a*



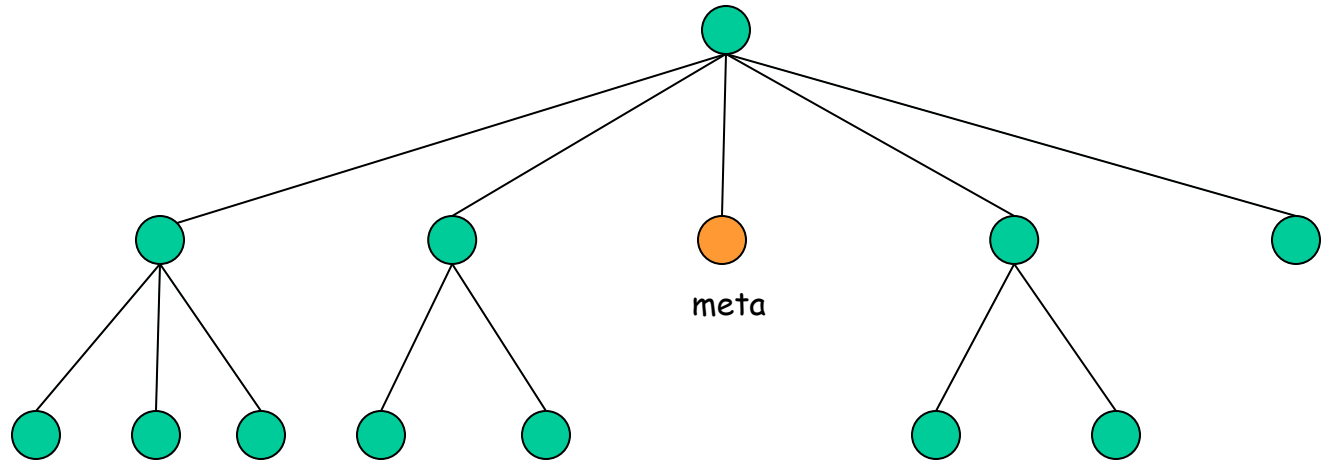
$$b^* = 2$$

# Búsqueda informada

---

## Heurísticas y desempeño

Calcule  $b^*$



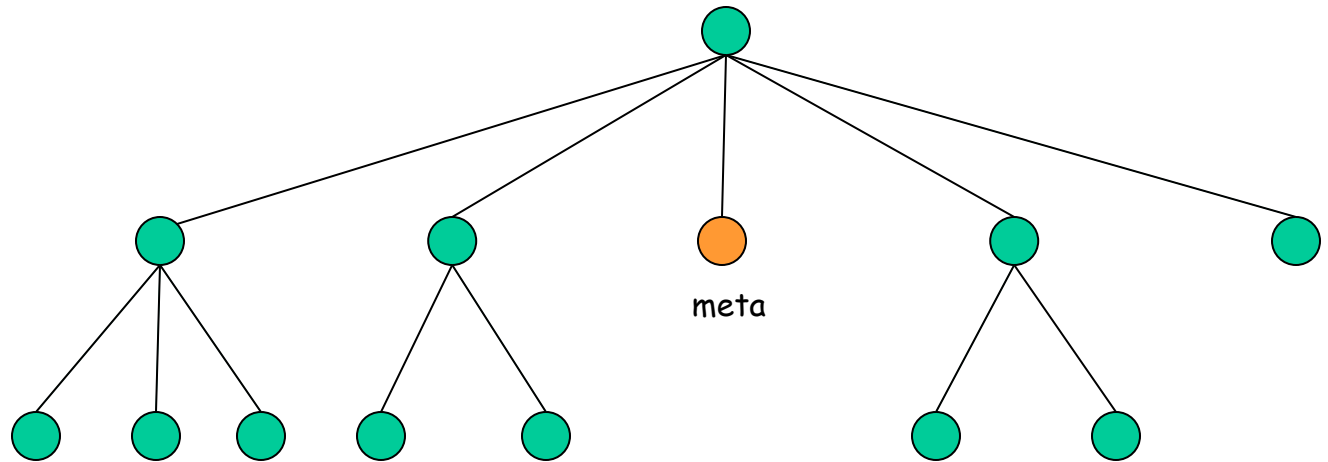
# Búsqueda informada

---

## Heurísticas y desempeño

$$N = 13$$

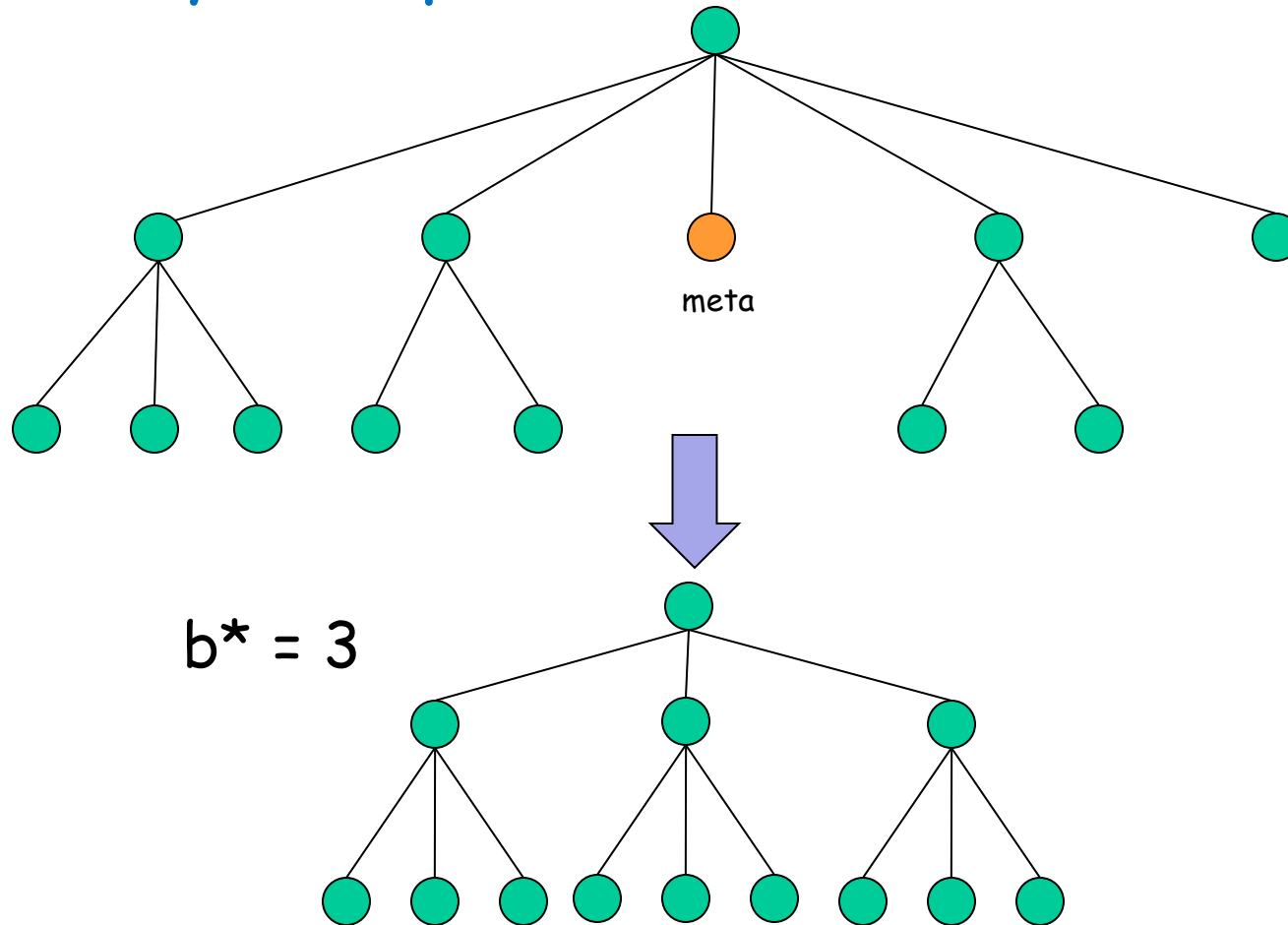
$$d = 2$$



$$1 + b^* + (b^*)^2 = 13$$

# Búsqueda informada

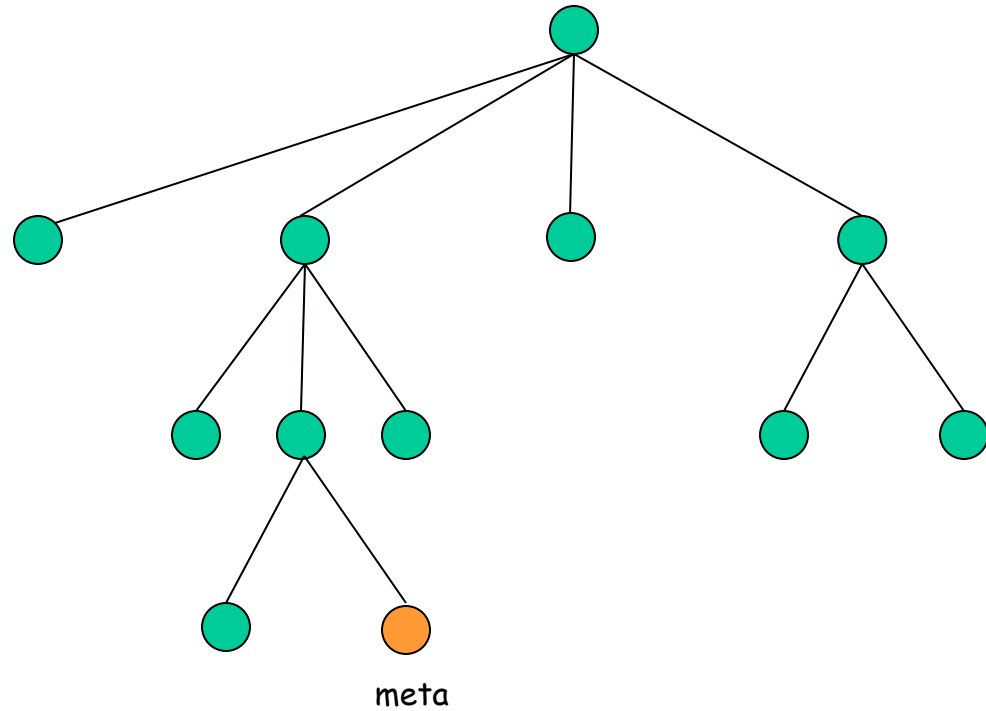
## Heurísticas y desempeño



# Búsqueda informada

---

## Heurísticas y desempeño



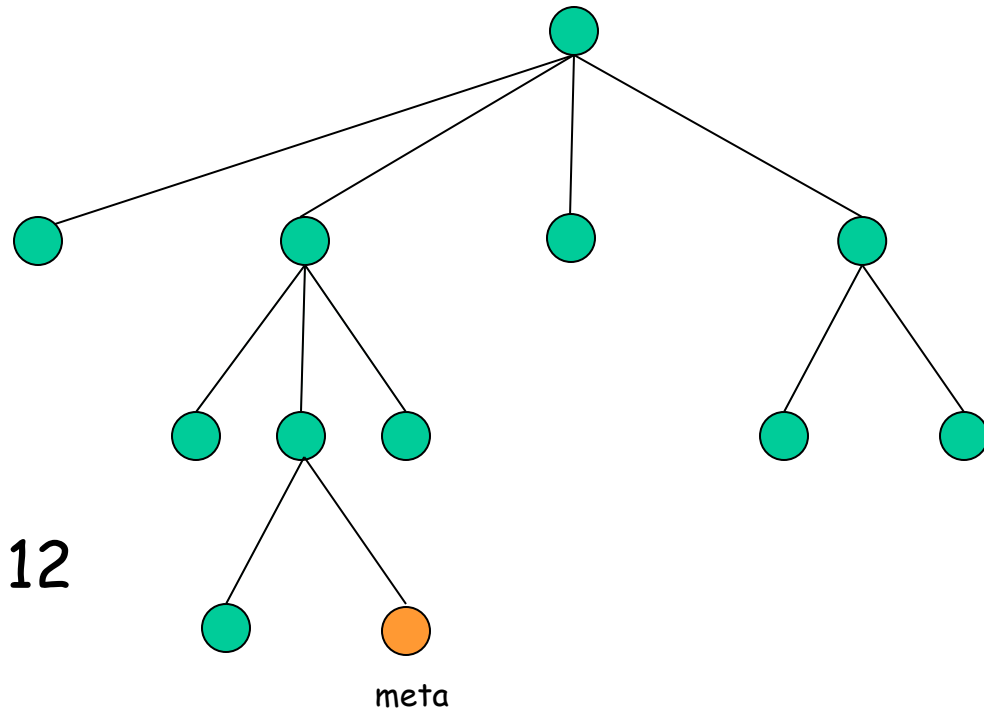
# Búsqueda informada

# Heurísticas y desempeño

$N = 12$

$$d = 3$$

$$1 + b^* + (b^*)^2 + (b^*)^3 = 12$$



# Búsqueda informada

---

## Heurísticas y desempeño

- $b^*$  es el factor de ramificación de  $A^*$  y se conoce como **factor de ramificación efectiva**
- Si en total el árbol generado tiene  $N$  nodos ( $d$  es la profundidad del árbol generado), se tendría que:

$$1 + b^* + (b^*)^2 + \dots + (b^*)^d = N$$

# Búsqueda informada

---

## Heurísticas y desempeño

- $b^*$  es el factor de ramificación de  $A^*$  y se conoce como **factor de ramificación efectiva**
- Si en total el árbol generado tiene  $N$  nodos ( $d$  es la profundidad del árbol generado), se tendría que:

$$1 + b^* + (b^*)^2 + \dots + (b^*)^d = N$$

¿Qué valor para el factor de ramificación cree que es más eficiente en una búsqueda?



# Búsqueda informada

---

## Heurísticas y desempeño

- En una heurística bien diseñada,  $b^*$  se aproxima a 1

# Búsqueda informada

## 8-puzzle

$h_1(n)$ : cantidad de piezas ubicadas en el lugar incorrecto

$h_2(n)$ : suma de las distancias de Manhattan

$d$	Nodos expandidos		Factor de ramificación efectivo	
	$A^*(h_1)$	$A^*(h_2)$	$A^*(h_1)$	$A^*(h_2)$
2	6	6	1.79	1.79
4	13	12	1.48	1.45
6	20	18	1.34	1.30
8	39	25	1.33	1.24
10	93	39	1.38	1.22
12	227	73	1.42	1.24
14	539	113	1.44	1.23
16	1,301	211	1.45	1.25
18	3,056	363	1.46	1.26
20	7,276	676	1.47	1.27
22	18,094	1,219	1.48	1.28
24	39,135	1,641	1.48	1.26

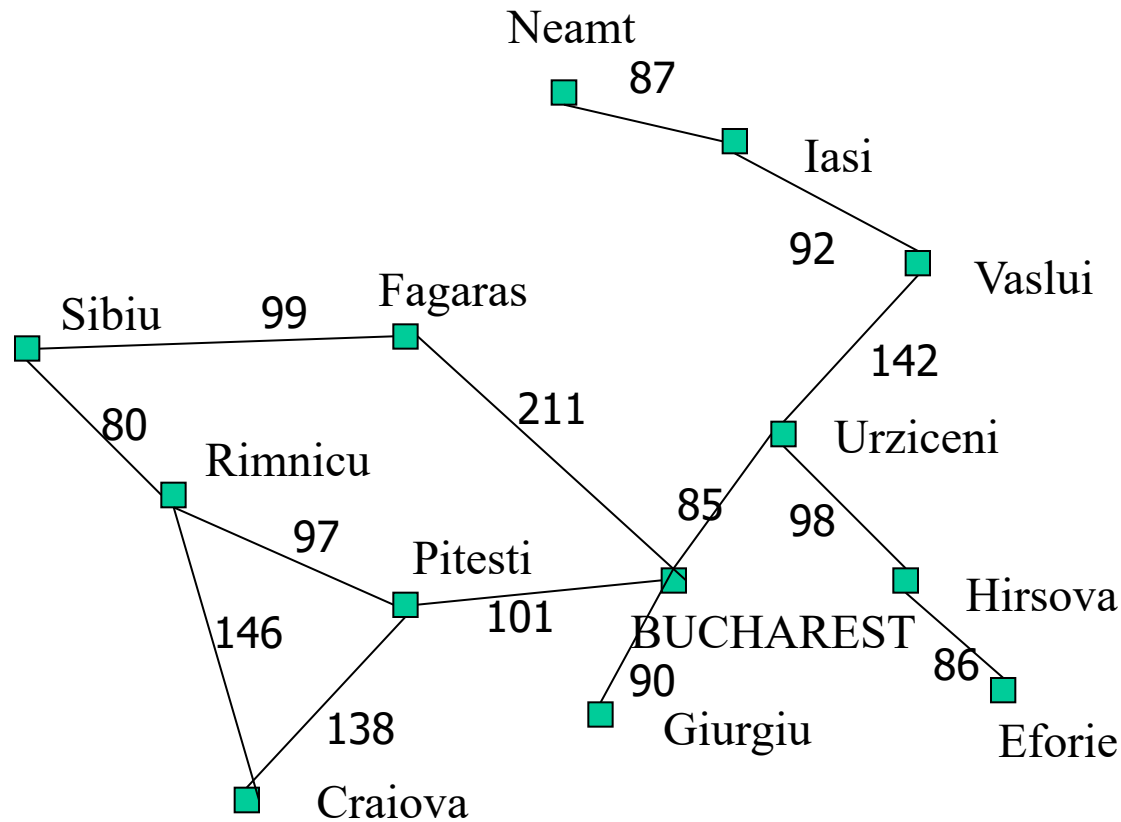
# Búsqueda informada

---

## Heurísticas y desempeño

- Si para todo nodo  $n$ ,  $h_2(n) \geq h_1(n)$ , se dice que  $h_2$  domina a  $h_1$
- En general, conviene más emplear una función heurística con valores más altos, siempre y cuando no dé lugar a una sobreestimación

# Búsqueda informada



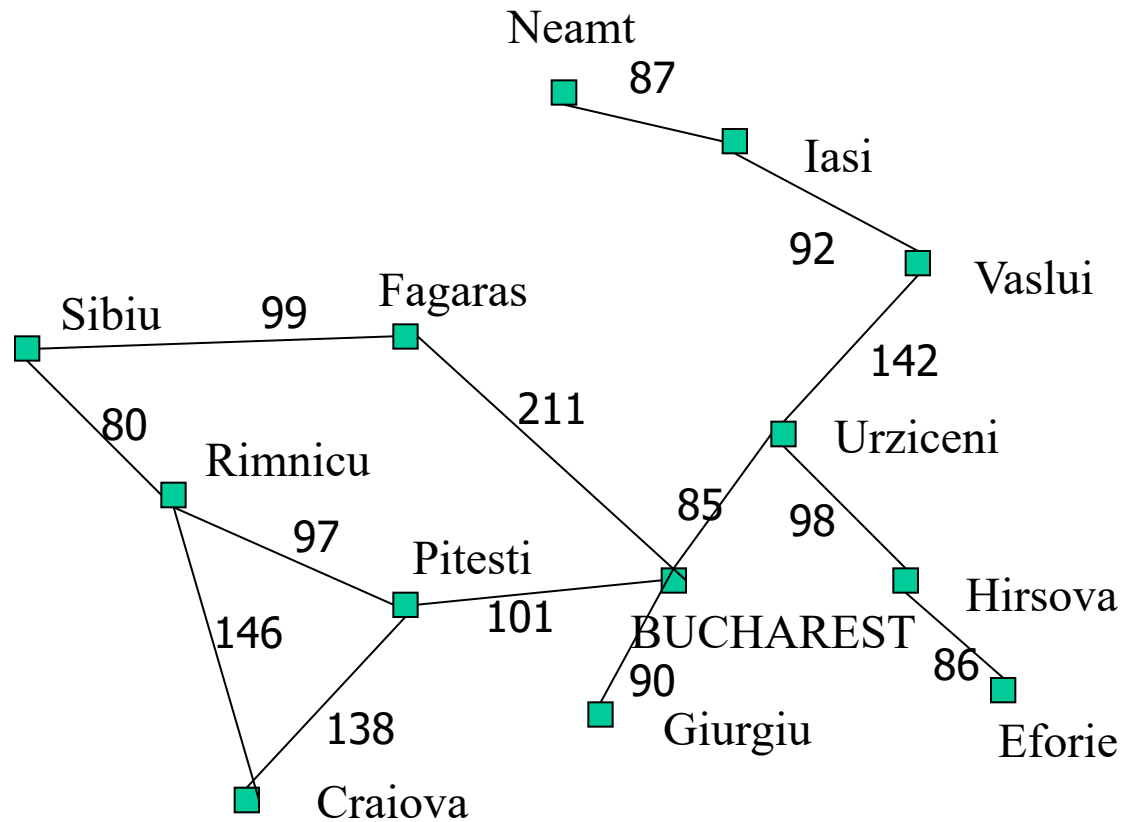
**Distancia en  
línea recta a  
Bucarest:**

Arad	366
Bucarest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

**Distancia mínima  
Bucarest:**

Arad	1
Bucarest	0
Craiova	1
Dobreta	1
Eforie	1
Fagaras	1
Giurgiu	1
Hirsova	1
Iasi	1
Lugoj	1
Mehadia	1
Neamt	1
Oradea	1
Pitesti	1
Rimnicu	1
Sibiu	1
Timisoara	1
Urziceni	1
Vaslui	1
Zerind	1

# Búsqueda informada



$h_1$  domina a  $h_2$

Distancia en  
línea recta a  
Bucarest:

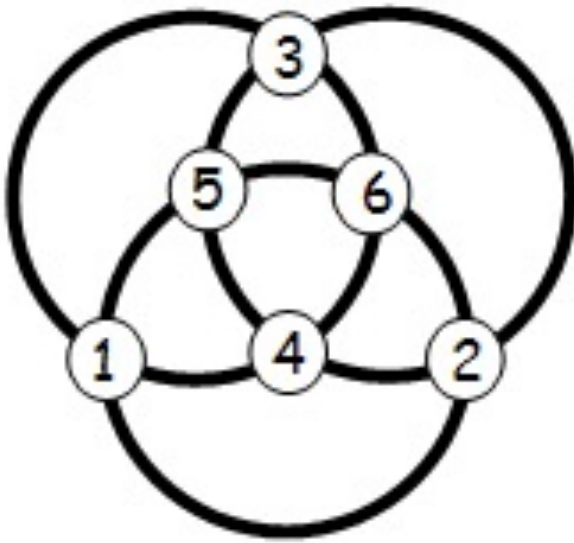
Arad	366
Bucarest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Distancia mínima  
Bucarest:

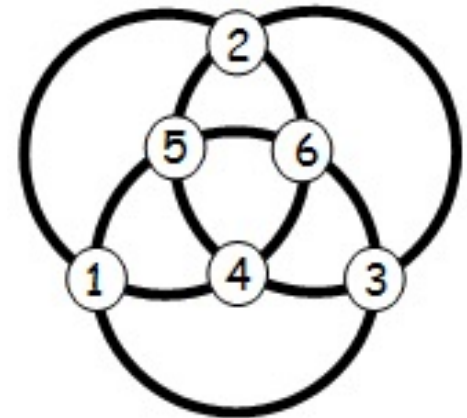
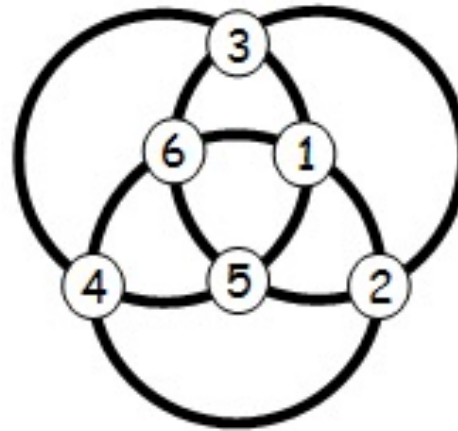
Arad	1
Bucarest	0
Craiova	1
Dobreta	1
Eforie	1
Fagaras	1
Giurgiu	1
Hirsova	1
Iasi	1
Lugoj	1
Mehadia	1
Neamt	1
Oradea	1
Pitesti	1
Rimnicu	1
Sibiu	1
Timisoara	1
Urziceni	1
Vaslui	1
Zerind	1

# Búsqueda informada

## Heurísticas dominantes



meta

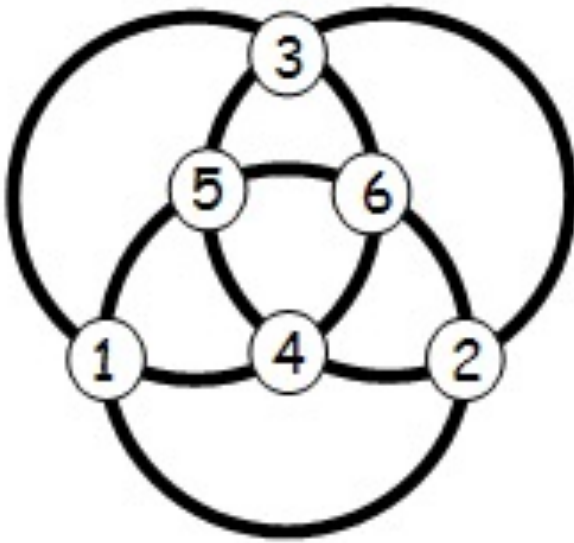


$$h_1(n) = \frac{\text{cantidad de aros que no suman 14}}{2}$$

$$h_2(n) = \frac{\text{cantidad de aros que no suman 14}}{100}$$

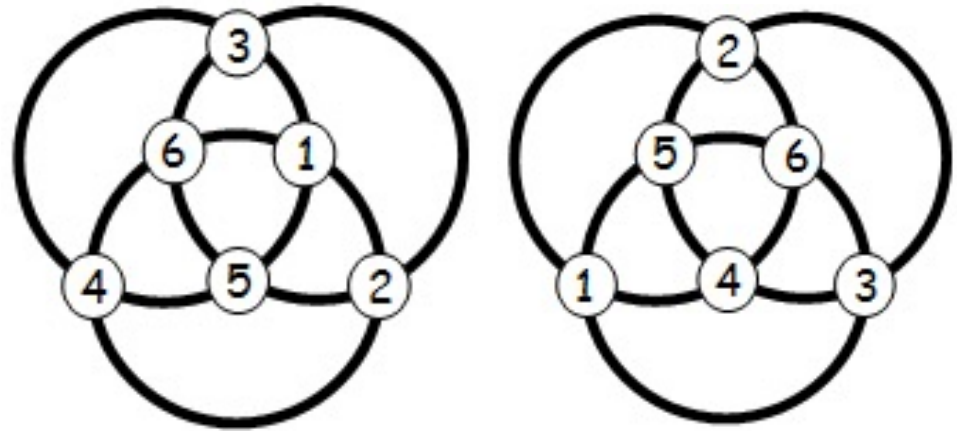
# Búsqueda informada

## Heurísticas dominantes



meta

$h_1$  domina a  $h_2$

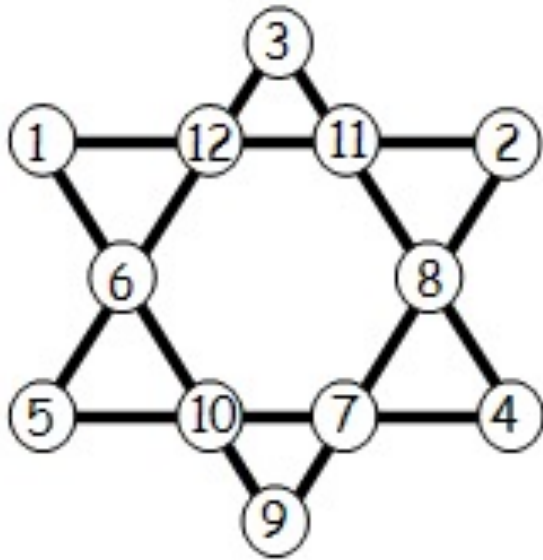


$$h_1(n) = \frac{\text{cantidad de aros que no suman 14}}{2}$$

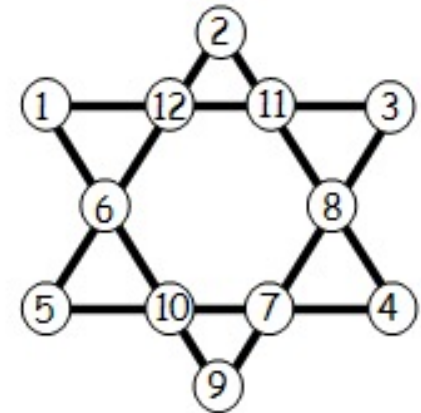
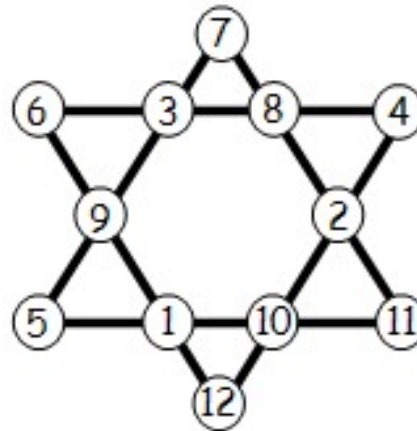
$$h_2(n) = \frac{\text{cantidad de aros que no suman 14}}{100}$$

# Búsqueda informada

## Heurísticas dominantes



meta



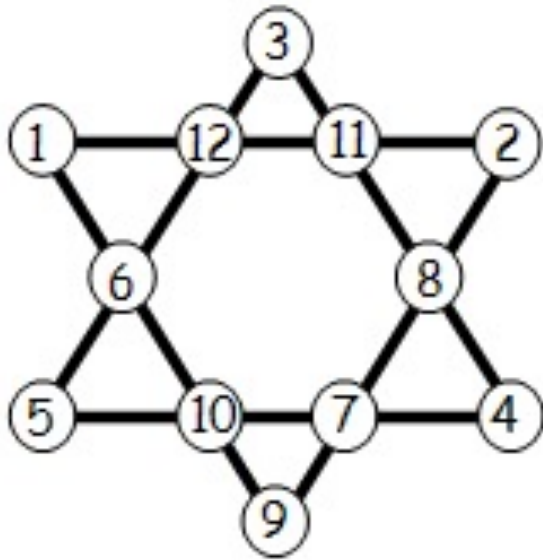
$$h_1(n) = \frac{\text{cantidad de lados que no suman 26}}{4}$$

$$h_2(n) = \frac{\text{cantidad de lados que no suman 26}}{5}$$



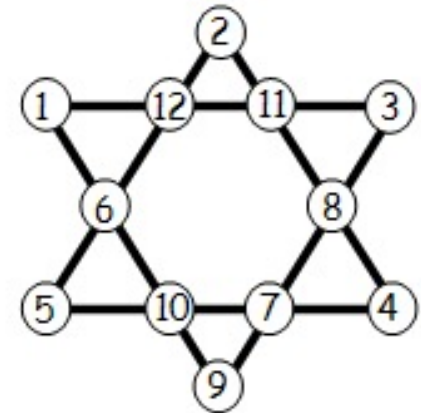
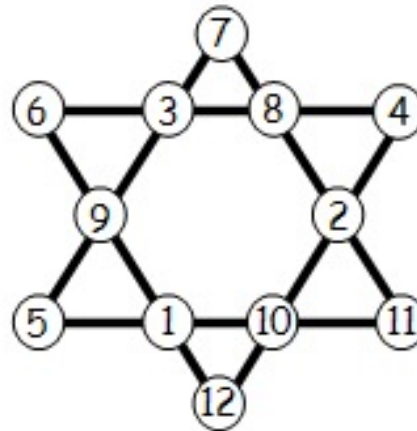
# Búsqueda informada

## Heurísticas dominantes



meta

$h_1$  domina a  $h_2$



$$h_1(n) = \frac{\text{cantidad de lados que no suman 26}}{4}$$

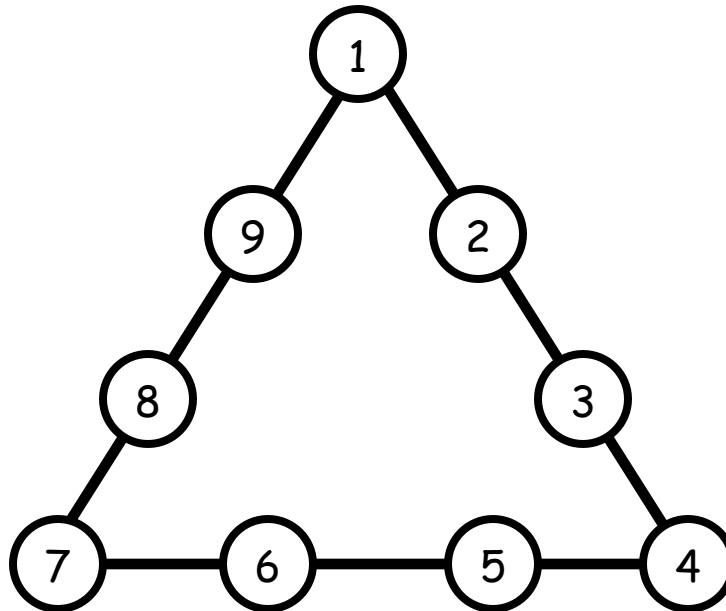
$$h_2(n) = \frac{\text{cantidad de lados que no suman 26}}{5}$$

# Búsqueda informada

---

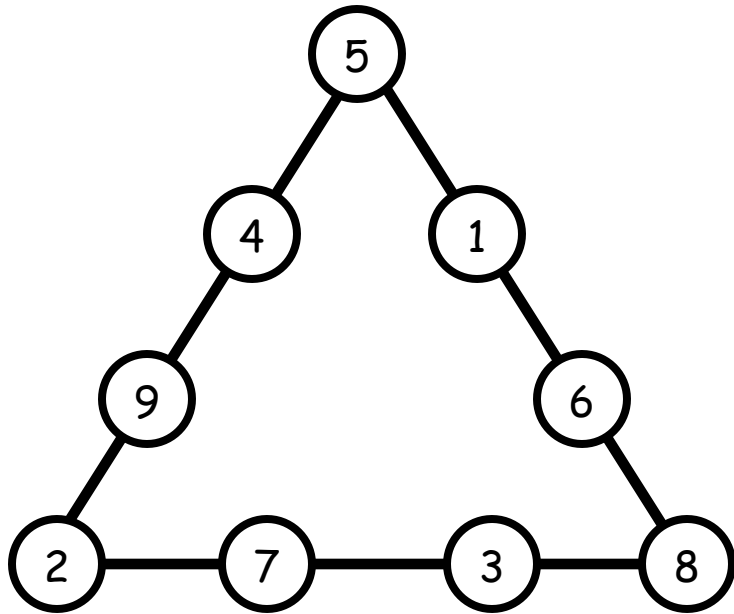
- Defina la heurística admisible más dominante

**El triángulo mágico.** Utilizando el operador *intercambio(a,b)* que cambia de posición los números a y b, colocarlos de tal forma que la suma sobre cada lado sea 20.

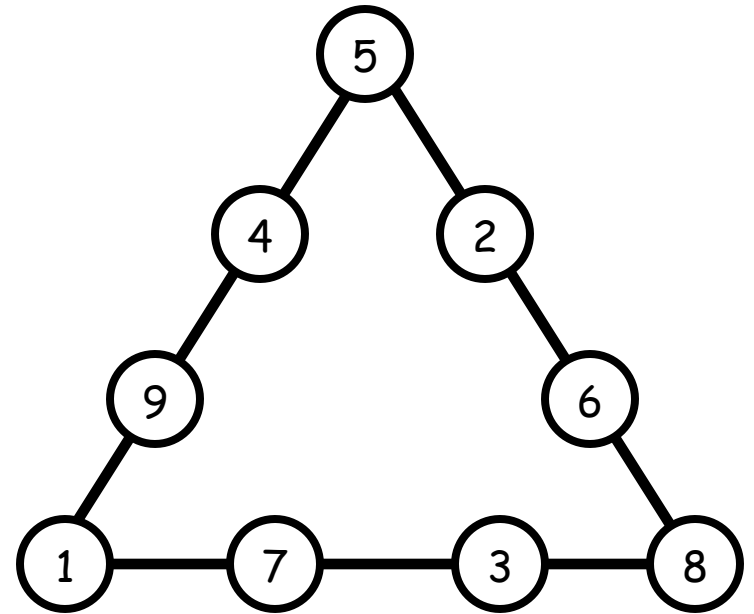


# Búsqueda informada

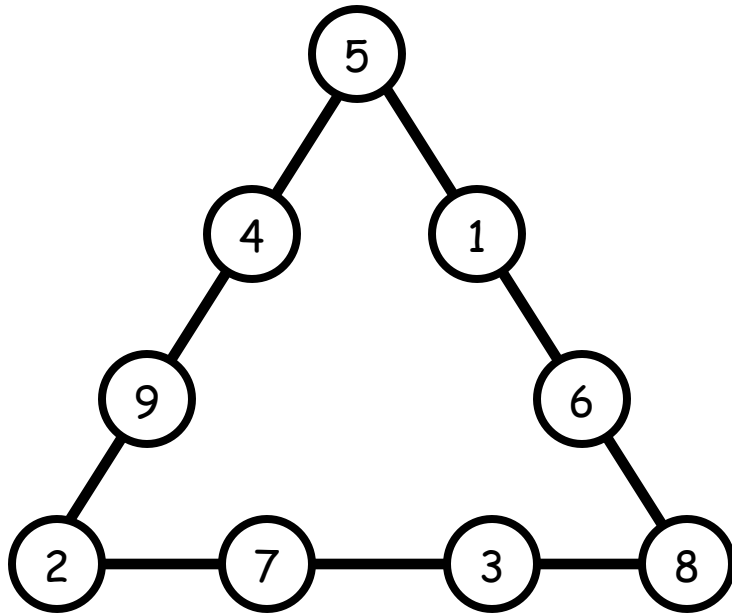
---



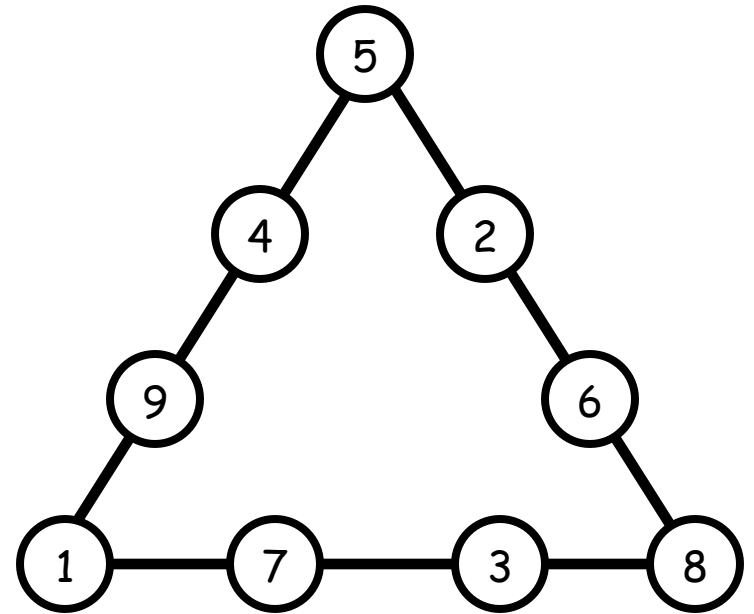
meta



# Búsqueda informada



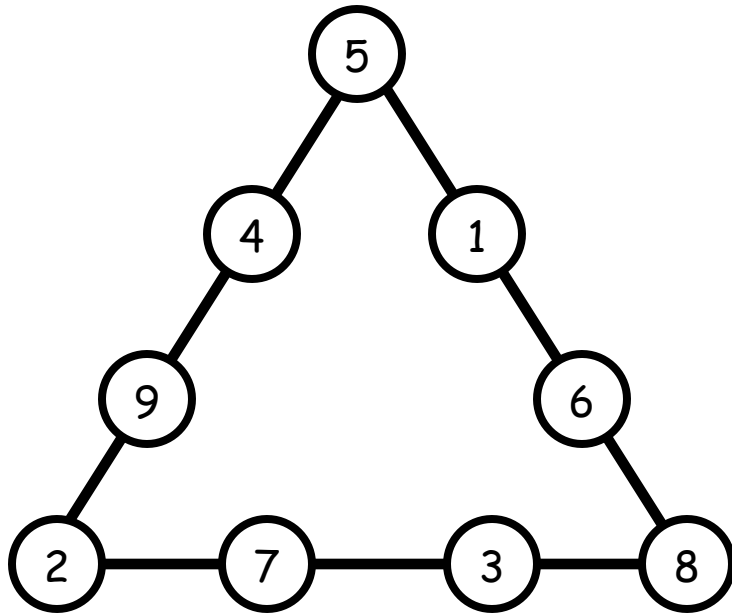
meta



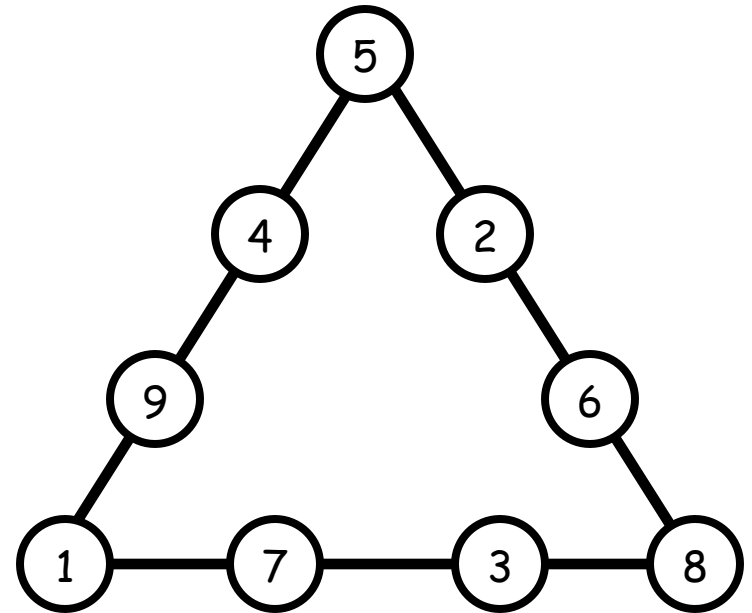
$h(n)$  = cantidad de lados que no suman 20

?

# Búsqueda informada



meta



$$h(n) = \frac{\text{cantidad de lados que no suman 20}}{3}$$

# Búsqueda informada

---

**Problema.** El cuadrado mágico

La suma de cada fila, columna y las diagonales debe ser 15. Los números en las celdas coloreadas no se pueden mover. En cada avance del juego se puede realizar la operación  $\text{intercambiar}(x,y)$  que permite intercambiar de posición los números  $x$  y  $y$ . Cada avance tiene costo de una unidad.

7	1	8
2	5	3
6	9	4

Defina la heurística  
admisible más dominante

# Búsqueda informada

Problema. El cuadrado mágico

6	1	8
7	5	3
2	9	4

meta

7	1	2
8	5	3
6	9	4

6	1	7
8	5	3
2	9	4

¿Cuál está más distante de la meta?

¿Cómo lo sabe?

# Búsqueda informada

Problema. El cuadrado mágico

6	1	8
7	5	3
2	9	4

meta

x	x	✓	x	x
x	7	1	2	
x	8	5	3	
x	6	9	4	

✓	x	✓	x	x
x	6	1	7	
x	8	5	3	
✓	2	9	4	

¿Cuál está más distante de la meta?

¿Cómo lo sabe?



# Búsqueda informada

Problema. El cuadrado mágico

6	1	8
7	5	3
2	9	4

meta

x	x	✓	x	x
x	7	1	2	
x	8	5	3	
x	6	9	4	

✓	x	✓	x	x
x	6	1	7	
x	8	5	3	
✓	2	9	4	

cantidad de filas, columnas y  
diagonales que no suman 15

$h(n) =$

?

# Búsqueda informada

Problema. El cuadrado mágico

6	1	8
7	5	3
2	9	4

meta

x	x	✓	x	x
x	7	1	2	
x	8	5	3	
x	6	9	4	

✓	x	✓	x	x
x	6	1	7	
x	8	5	3	
✓	2	9	4	





cantidad de filas, columnas y  
diagonales que no suman 15

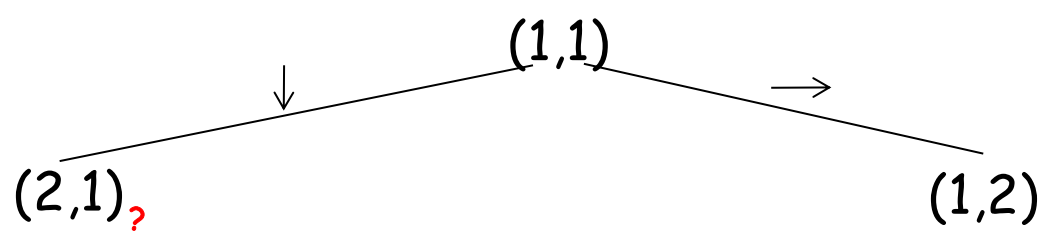
$h(n) =$





5

**4. Búsqueda informada. El ratón y las píldoras.** Considere el siguiente ambiente representado por una matriz de 4x6 donde se tiene un ratón que inicia en la posición (1,1) y debe llegar a la posición (4,6). Los operadores que puede aplicar el agente son arriba, abajo, izquierda, y derecha. Cada movimiento tiene costo 1. En el ambiente hay tres píldoras. Cuando el ratón llega a una casilla con píldora, sus dos siguientes movimientos son más rápidos y por lo tanto el costo de cada desplazamiento es  $1/2$ . Tenga en cuenta que cada píldora tiene efecto solamente en los dos movimientos que realiza el ratón justo después de tomar una píldora. Después de que pase el efecto de la píldora el costo de cada movimiento será 1 nuevamente. En el recorrido que realice el ratón se pueden usar varias píldoras.

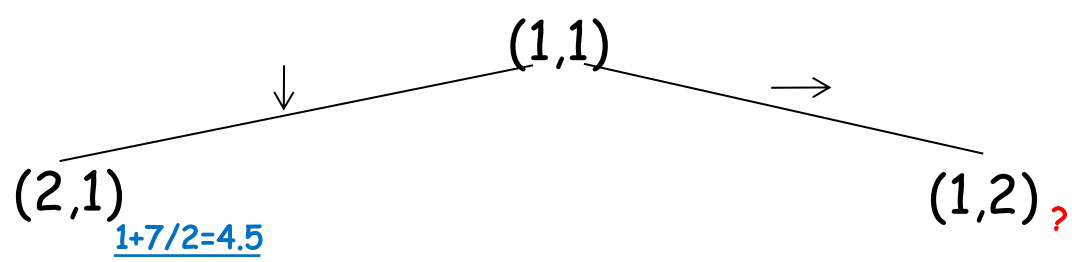
- **[I.L 1.5 (2pts)]** Defina una heurística que sea admisible para el ambiente que se muestra en la figura.
- **[I.L 1.6 (7pts)]** Muestre el árbol y la solución que se encuentra con A\*. Debe aplicar los operadores en el orden,  $\downarrow$ ,  $\rightarrow$ ,  $\uparrow$ ,  $\leftarrow$ . En cada nodo debe indicar los valores de g y h. Evite devolverse en todos los casos. Incluso cuando el ratón tome una píldora, evite que se devuelva.





	1	2	3	4	5	6
1						
2						
3						
4						Fin



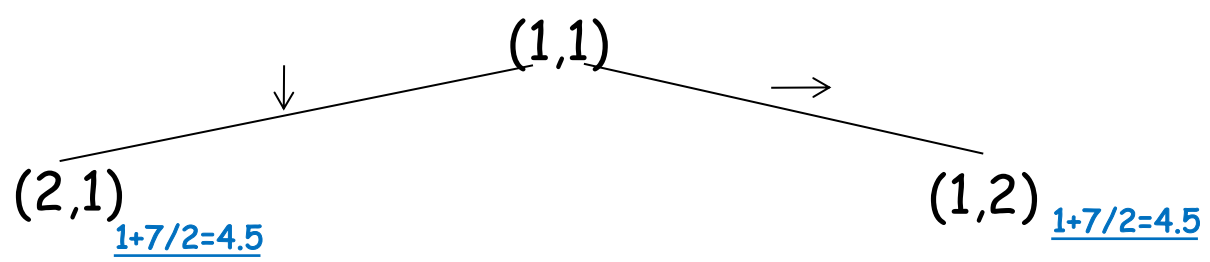
	1	2	3	4	5	6
1						
2						
3						
4						Fin





↓, →, ↑, ←.



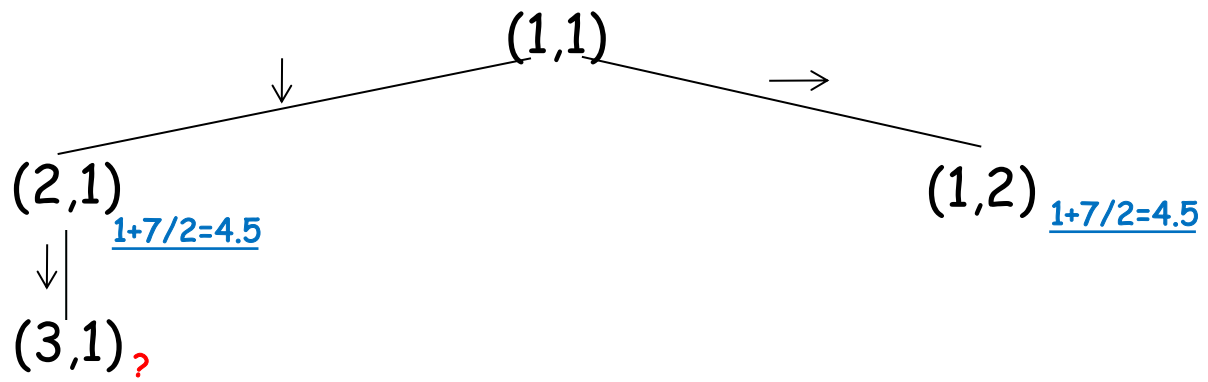
	1	2	3	4	5	6
1						
2						
3						
4						Fin





↓, →, ↑, ←.



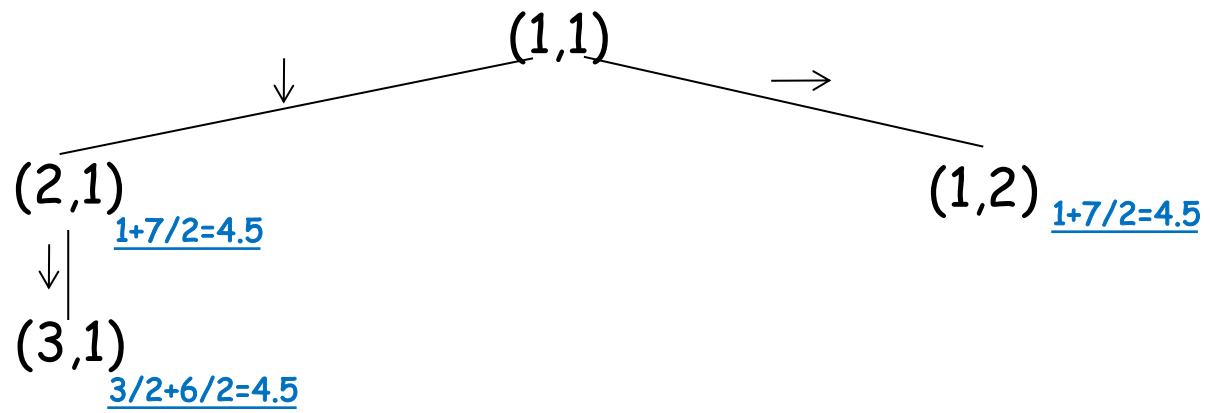
	1	2	3	4	5	6
1						
2						
3						
4						Fin





$\downarrow, \rightarrow, \uparrow, \leftarrow$



	1	2	3	4	5	6
1						
2						
3						
4						Fin

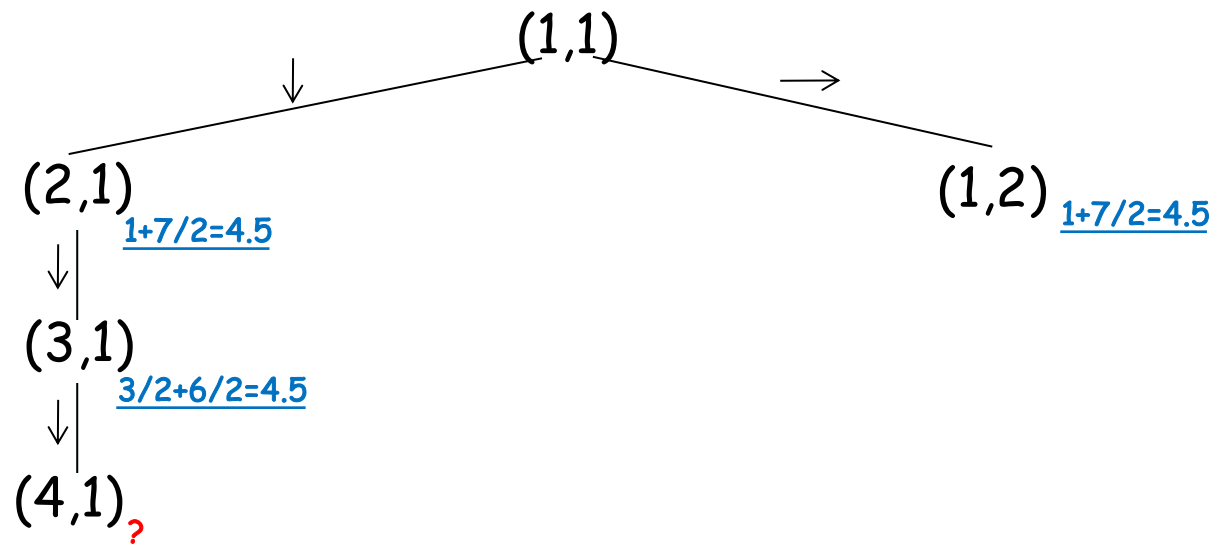
$\downarrow, \rightarrow, \uparrow, \leftarrow$







	1	2	3	4	5	6
1						
2						
3						
4						Fin

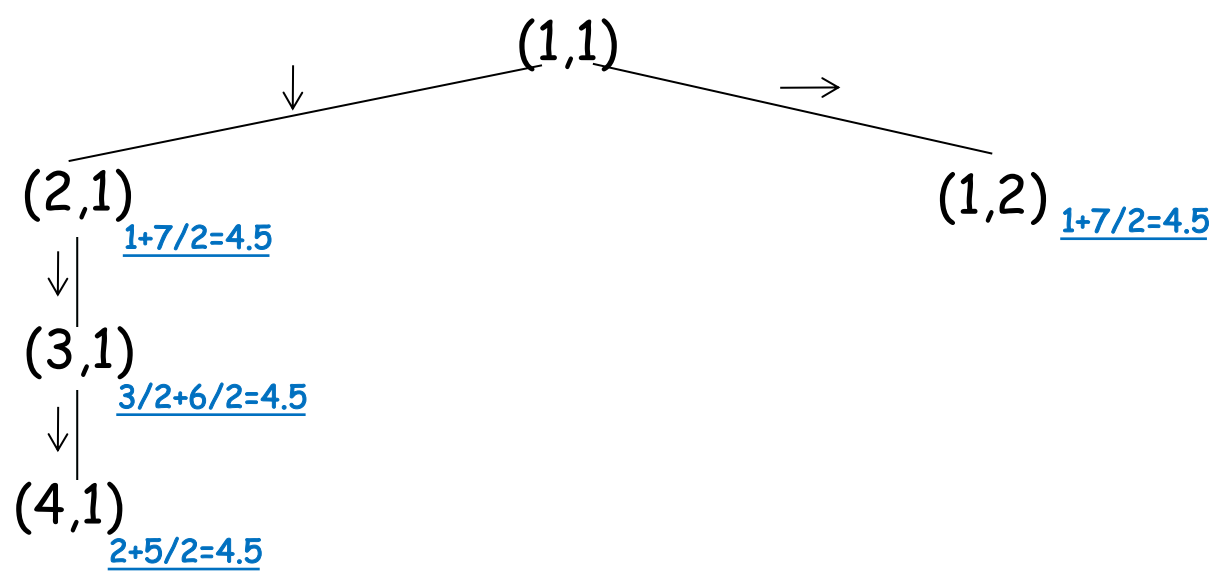
↓, →, ↑, ←.









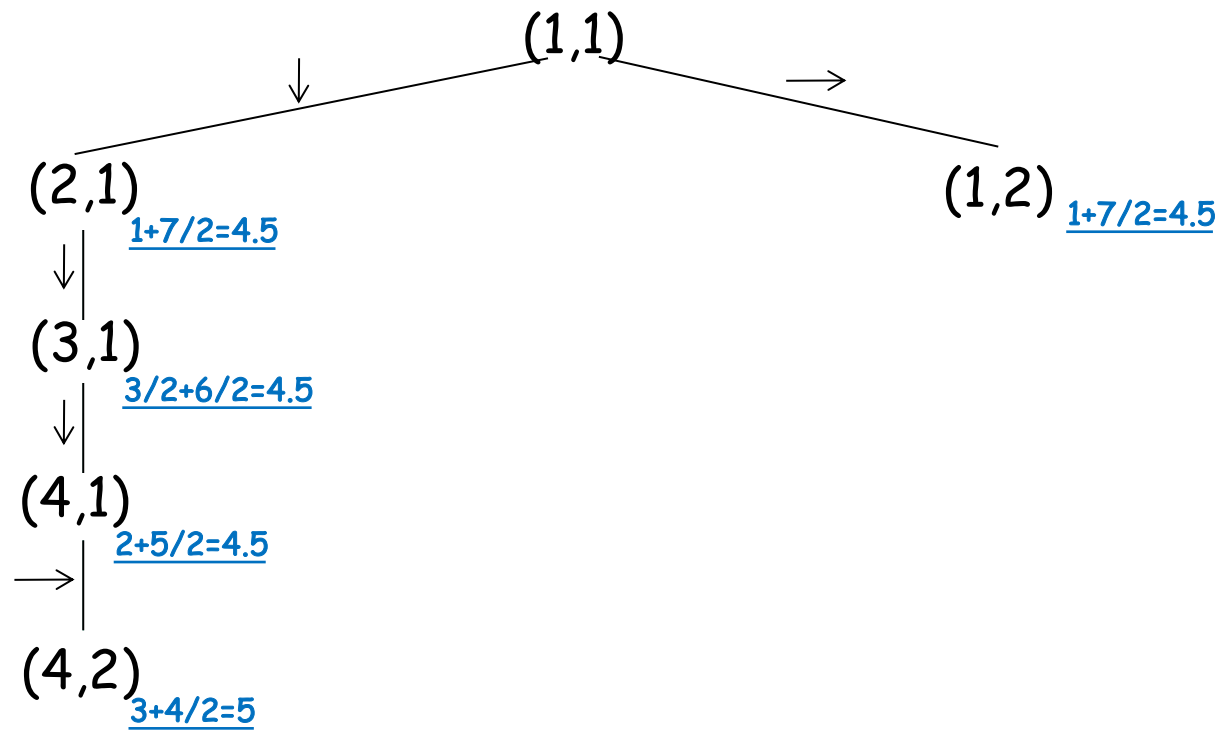
	1	2	3	4	5	6
1						
2						
3						
4						Fin





↓, →, ↑, ←.



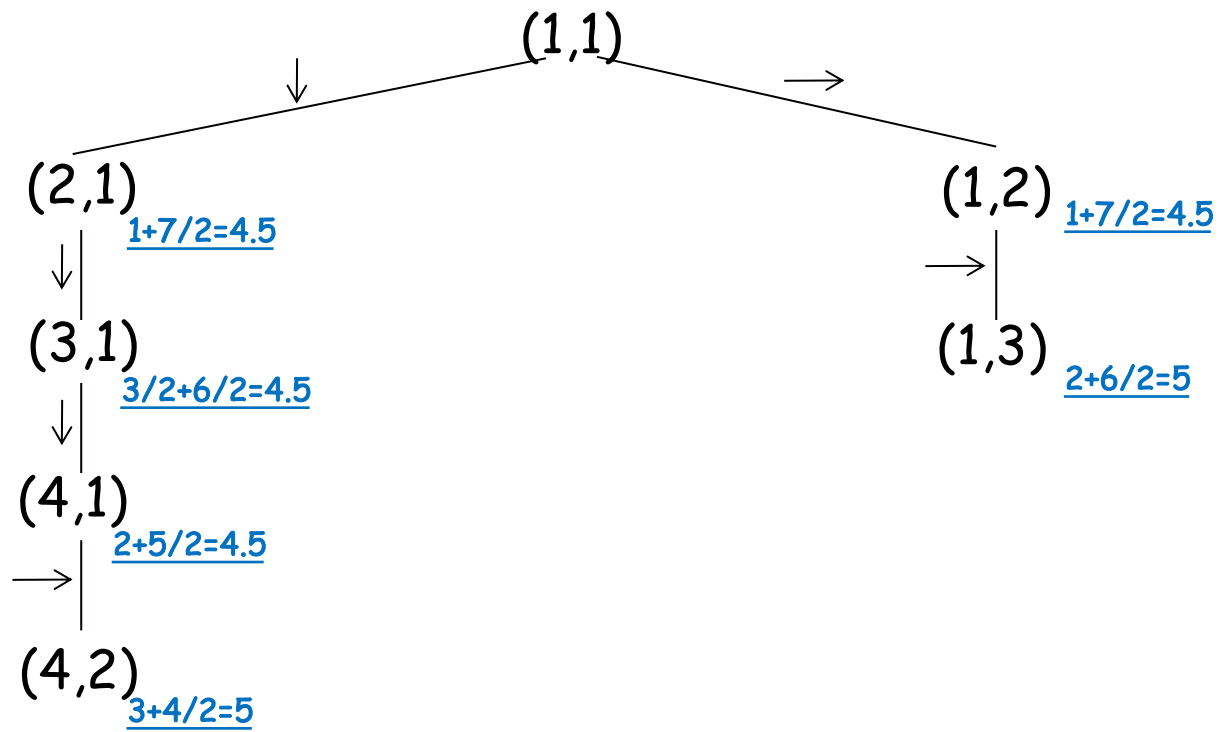
	1	2	3	4	5	6
1						
2						
3						
4						Fin





↓, →, ↑, ←.



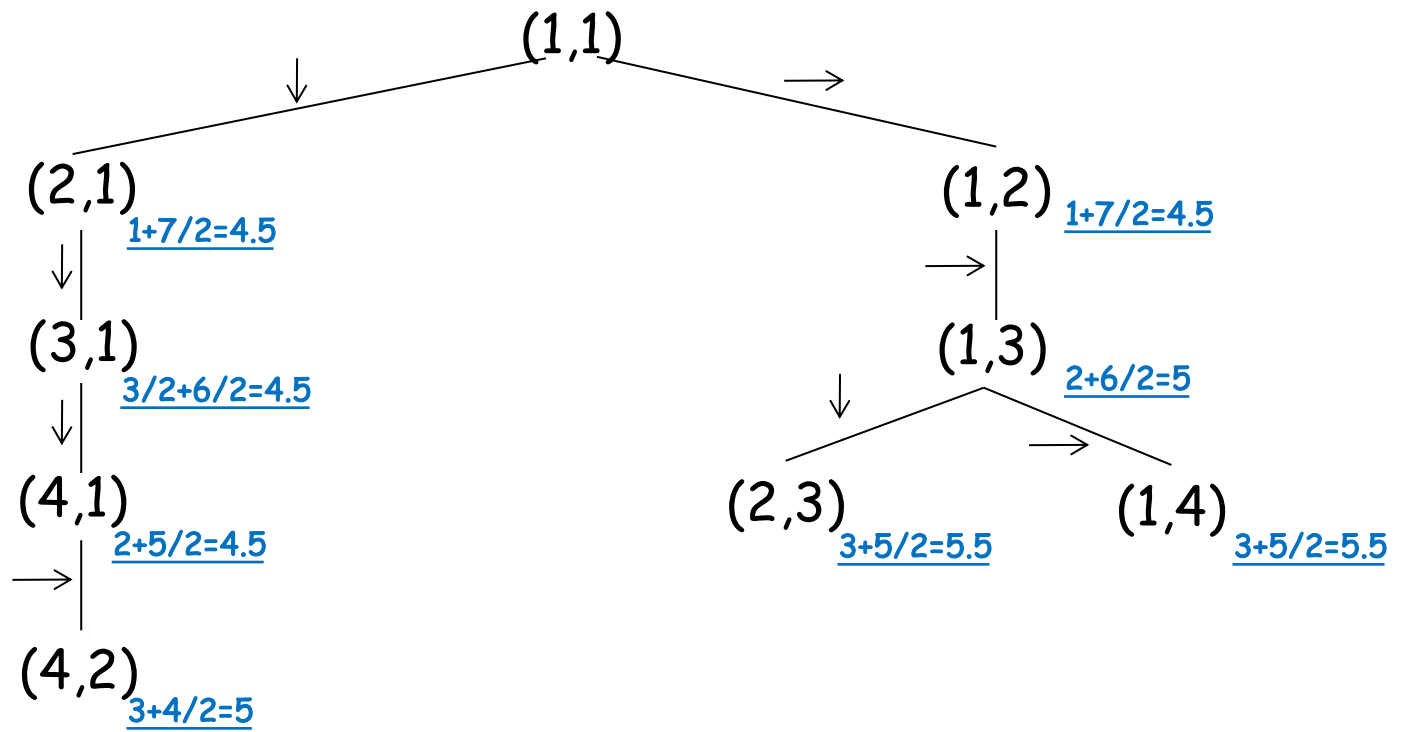
	1	2	3	4	5	6
1						
2						
3						
4						Fin





↓, →, ↑, ←.



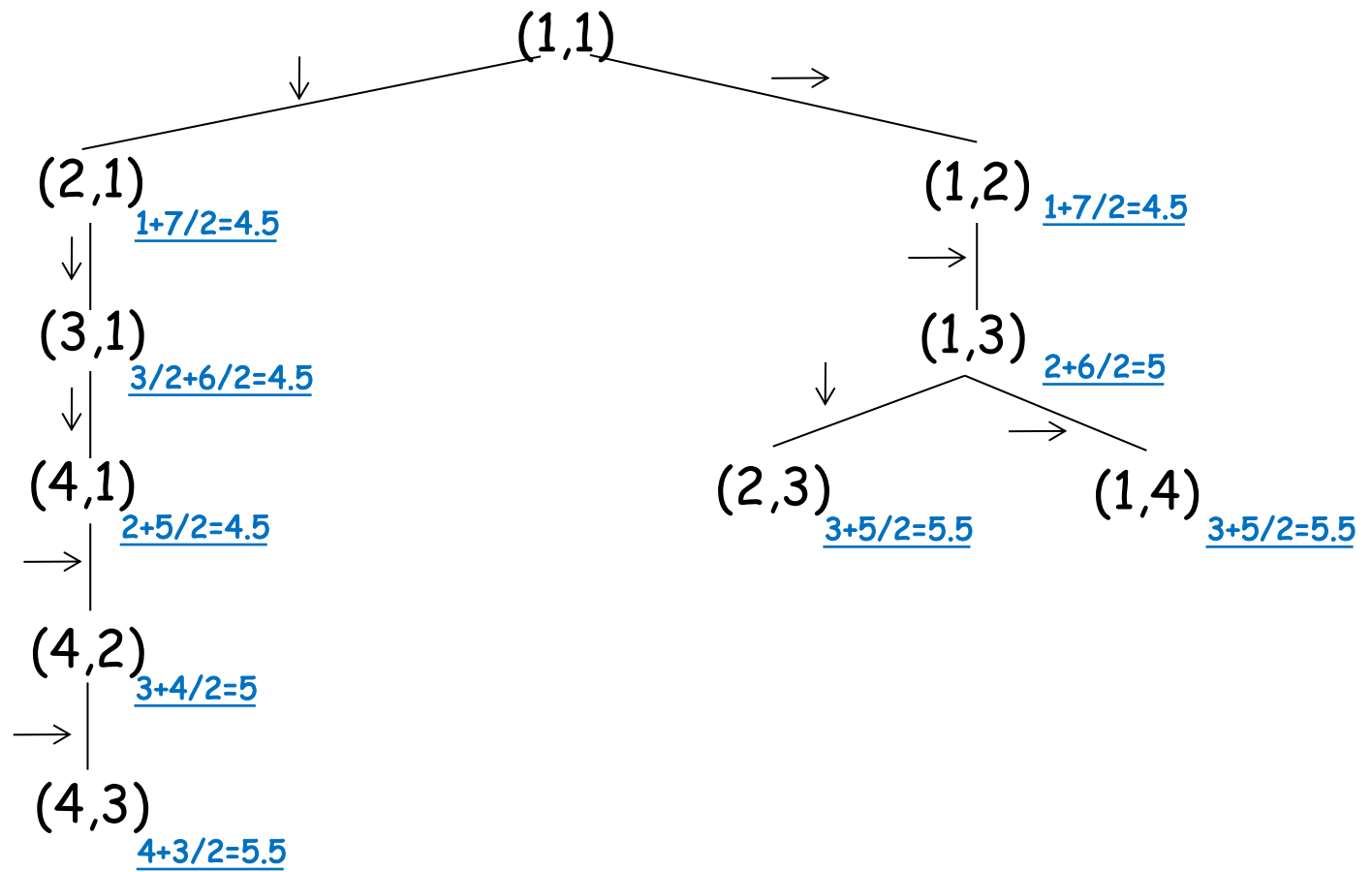
	1	2	3	4	5	6
1						
2						
3						
4						Fin





↓, →, ↑, ←.



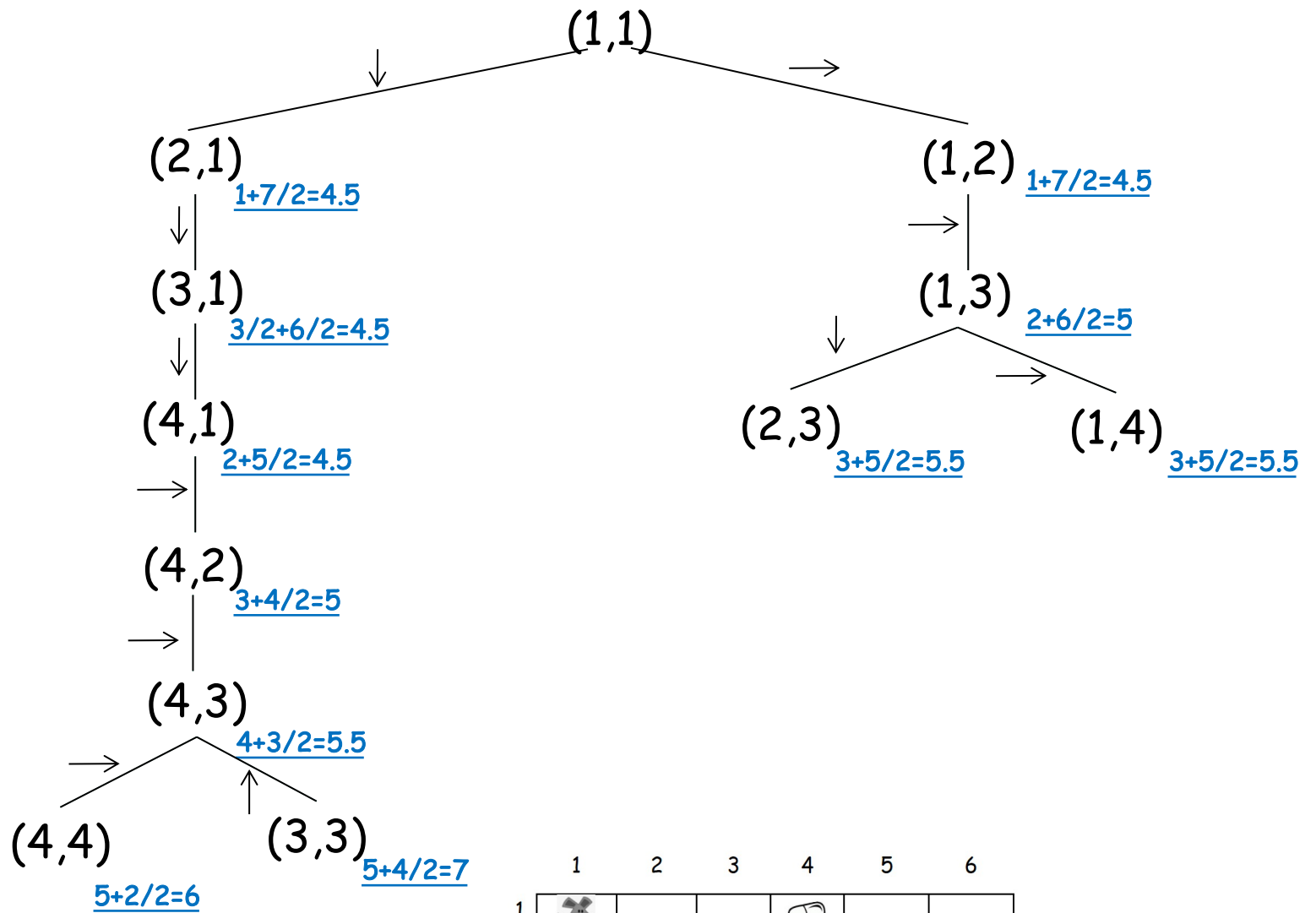
	1	2	3	4	5	6
1						
2						
3						
4						Fin





↓, →, ↑, ←.



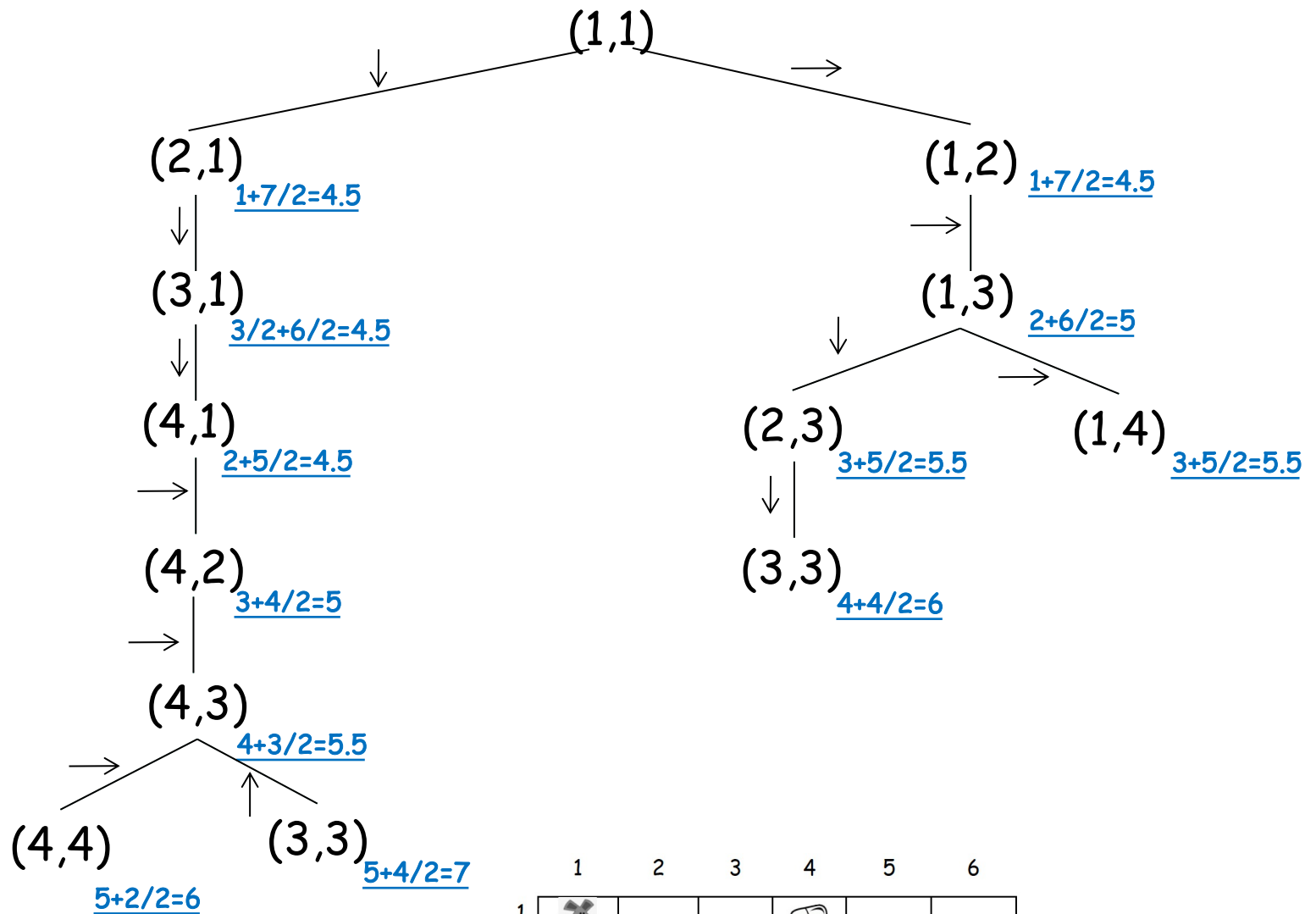
	1	2	3	4	5	6
1						
2						
3						
4						Fin





↓, →, ↑, ←.



	1	2	3	4	5	6
1						
2						
3						
4						Fin

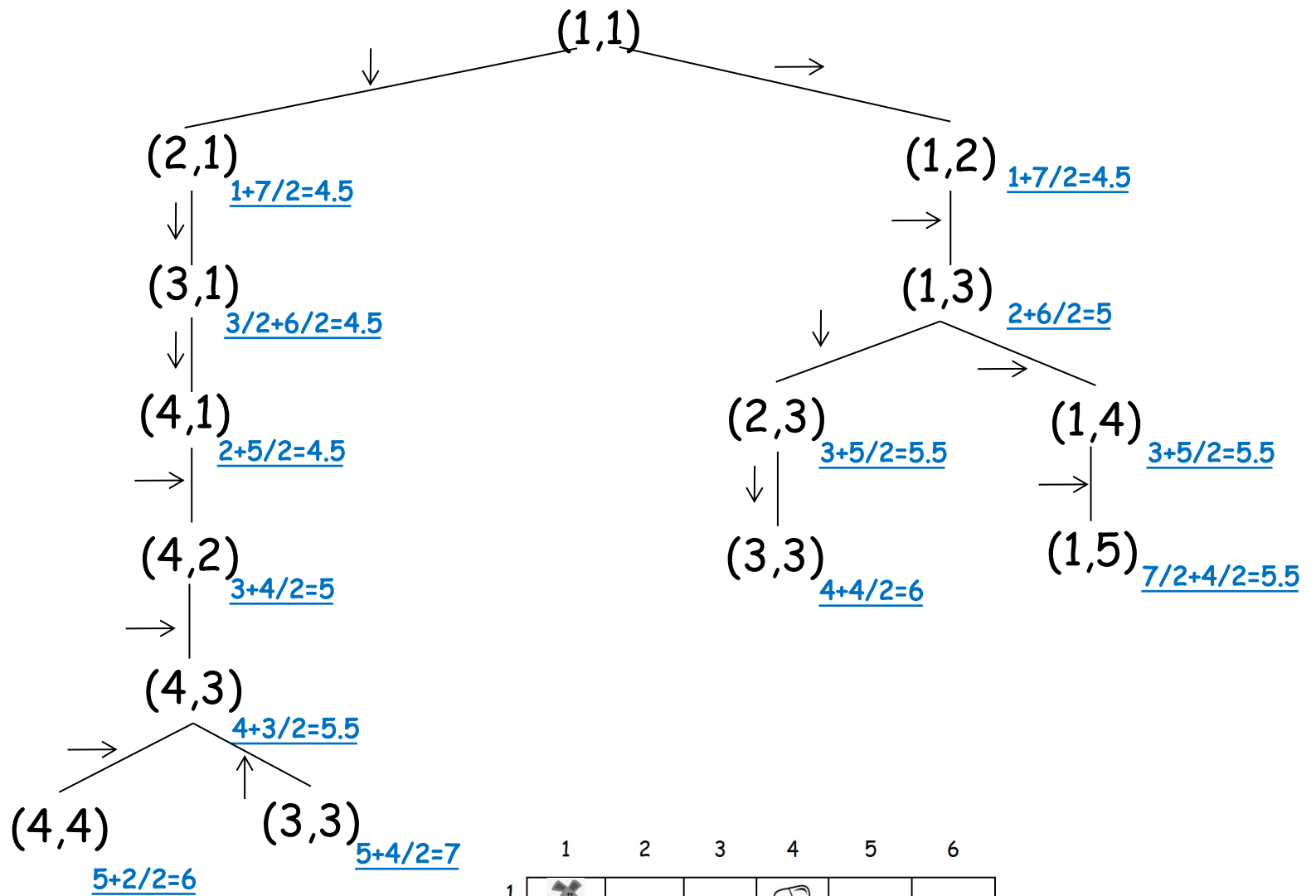
↓, →, ↑, ←.







	1	2	3	4	5	6
1						
2						
3						
4						Fin

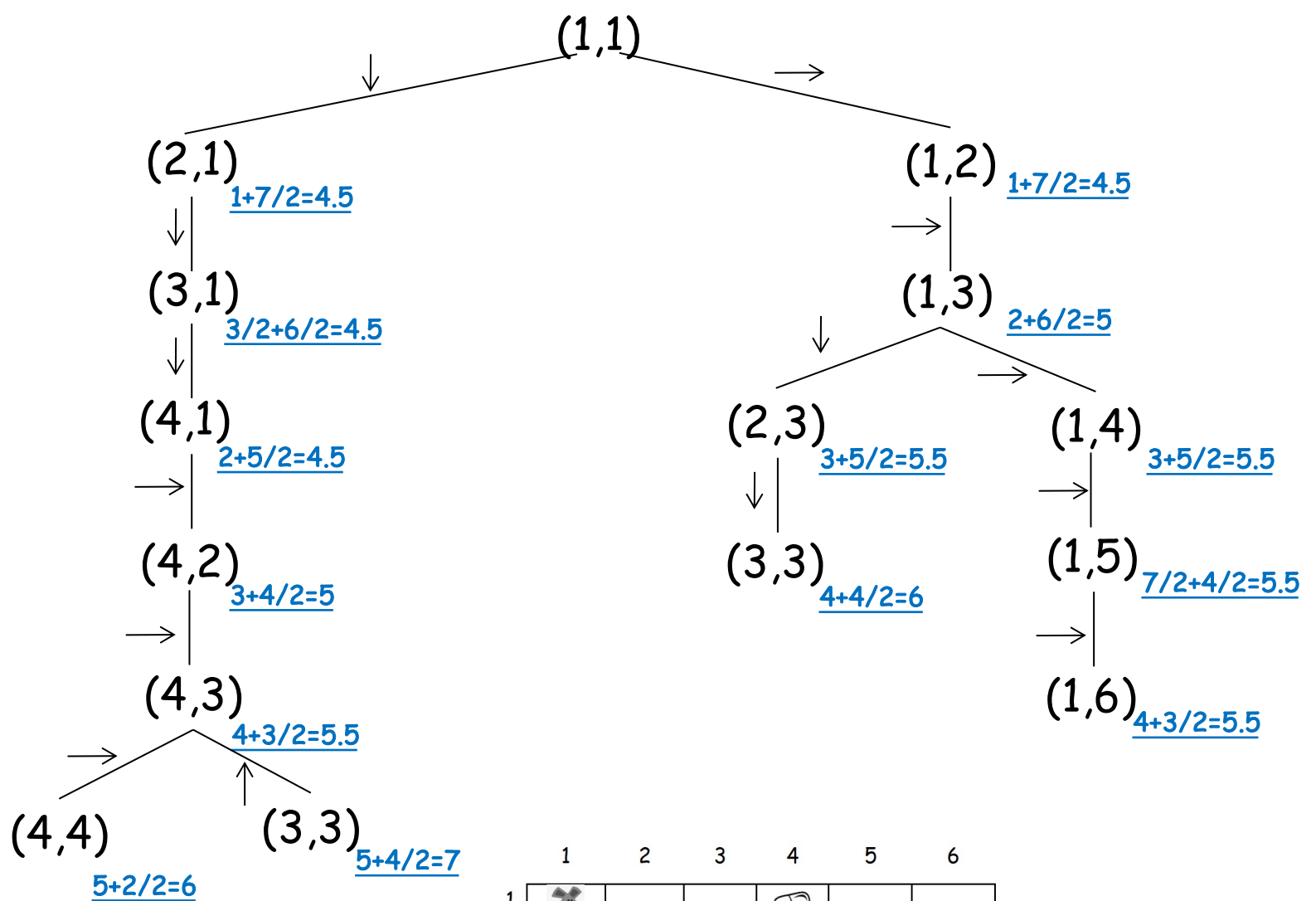
↓, →, ↑, ←.









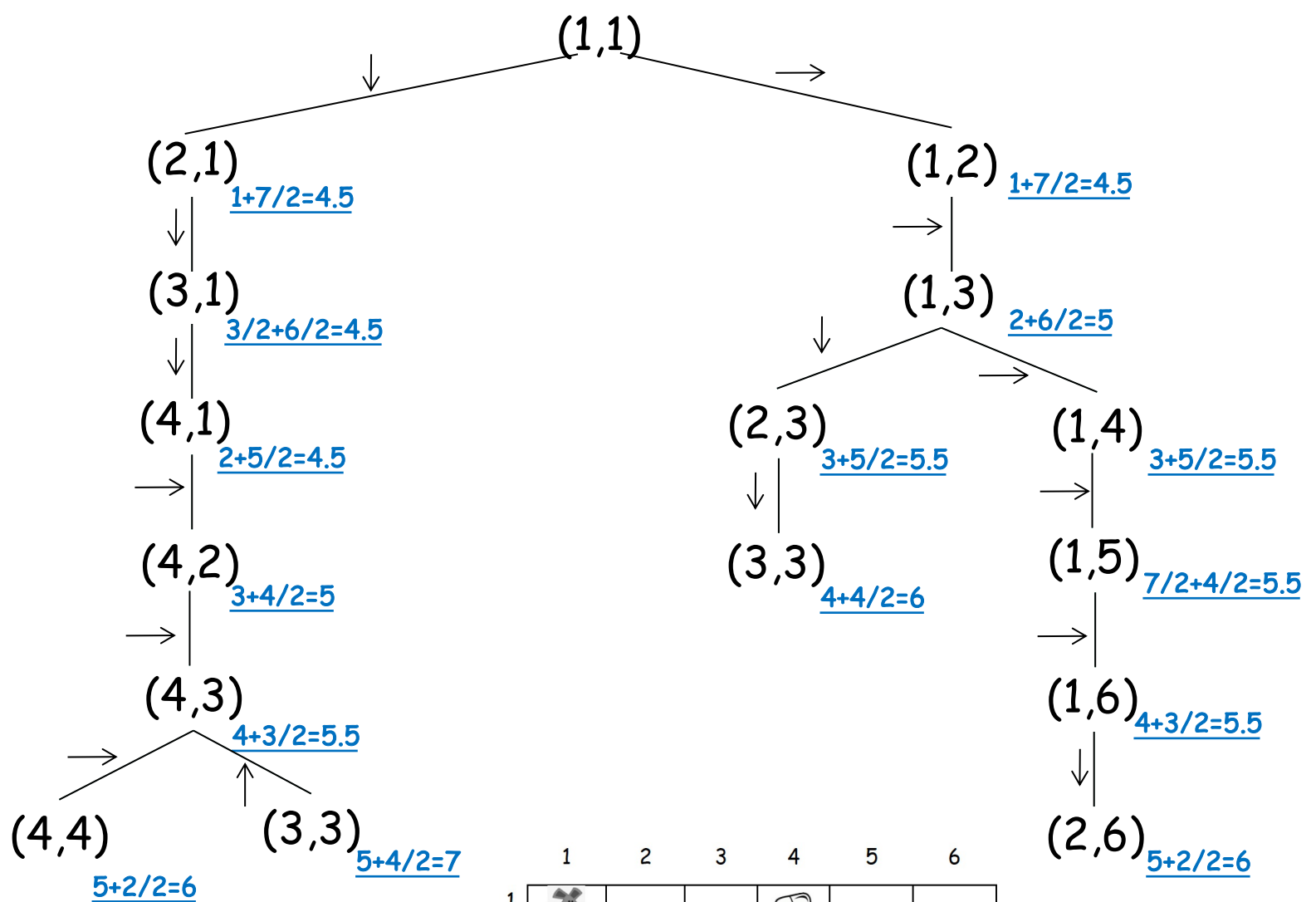
	1	2	3	4	5	6
1						
2						
3						
4						Fin





$\downarrow, \rightarrow, \uparrow, \leftarrow$



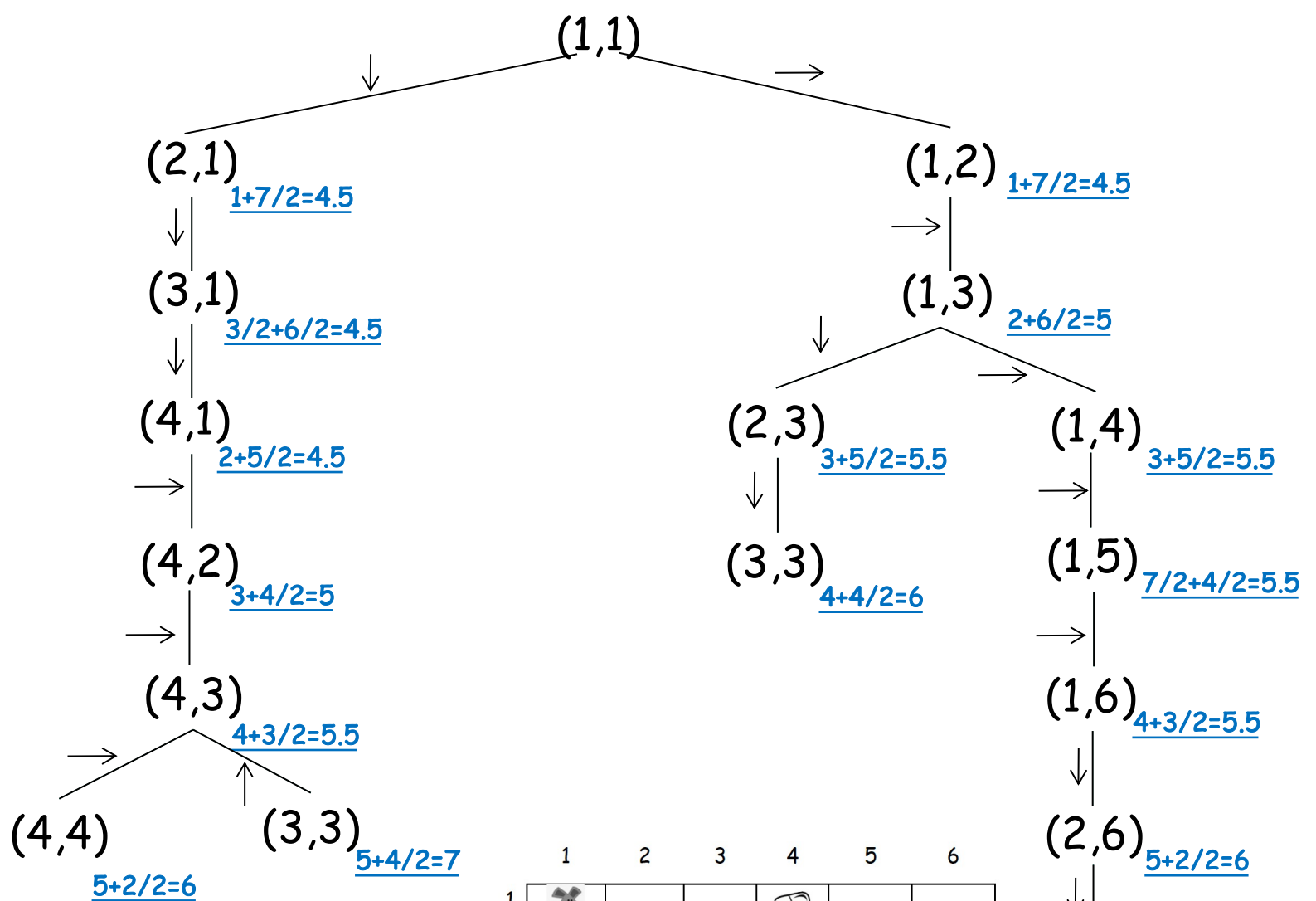
	1	2	3	4	5	6
1						
2						
3						
4						Fin





↓, →, ↑, ←.



	1	2	3	4	5	6
1						
2						
3						
4						Fin

↓, →, ↑, ←.



	1	2	3	4	5	6
1						
2						
3						
4						Fin

↓, →, ↑, ←.

