

Inteligencia Artificial

Oscar Bedoya

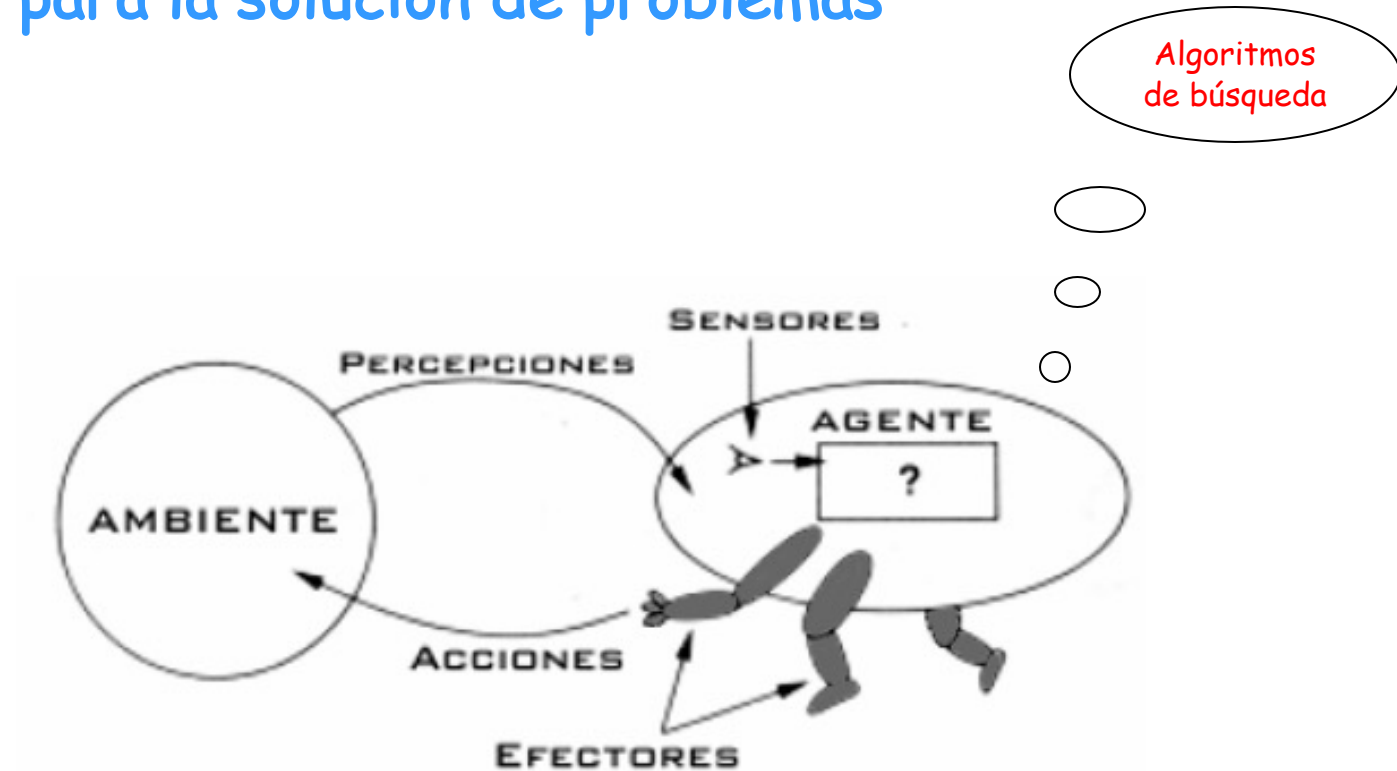
`oscar.bedoya@correounivalle.edu.co`

- * Agentes que resuelven problemas
- * Árbol de búsqueda
- * Algoritmos de búsqueda no informada



Búsqueda no informada

Agente para la solución de problemas



Búsqueda no informada

Cómo solucionar un problema

1. **Formulación del problema:** conjunto de estados, operadores, meta y costo asociado
2. **Búsqueda:** evaluar acciones posibles y decidir la más apropiada
3. **Ejecución:** una vez encontrada la solución, se procede a ejecutar la secuencia de acciones

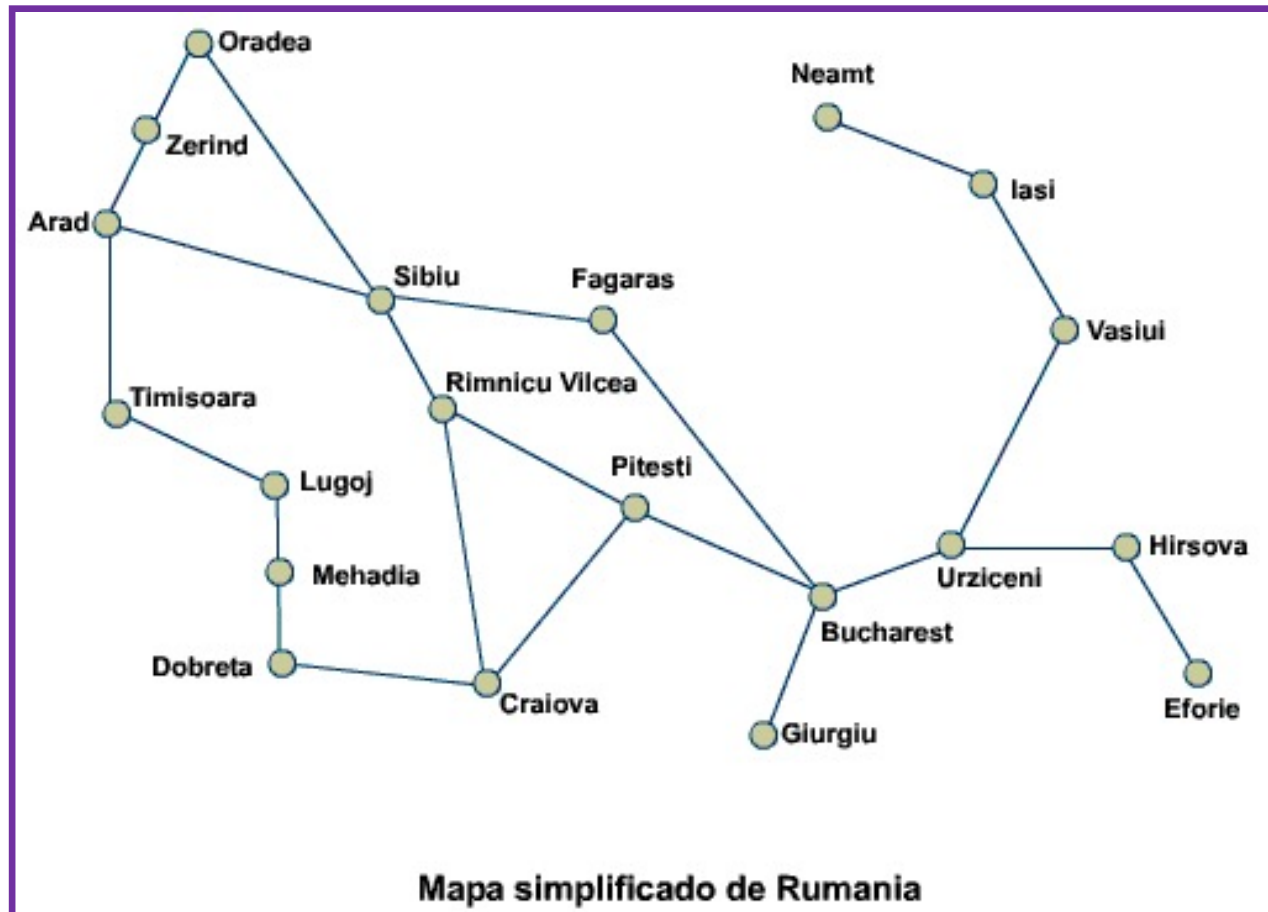
Búsqueda no informada

Problemas bien definidos

- Estados
- Operadores
- Prueba de meta
- Función costo de ruta

Búsqueda no informada

Problema: partiendo desde Arad llegar a Bucarest



Búsqueda no informada

Problemas bien definidos

- Estados
- Operadores
- Prueba de meta
- Función costo de ruta

Búsqueda no informada

Problemas bien definidos

- **Estados**
 - Cómo se representa el estado actual
 - Cuáles son los posibles estados
 - Estado inicial

Búsqueda no informada

Problemas bien definidos

- **Operadores**

Aplicado sobre un estado particular genera un conjunto de estados posibles

Búsqueda no informada

Problemas bien definidos

- Prueba de meta

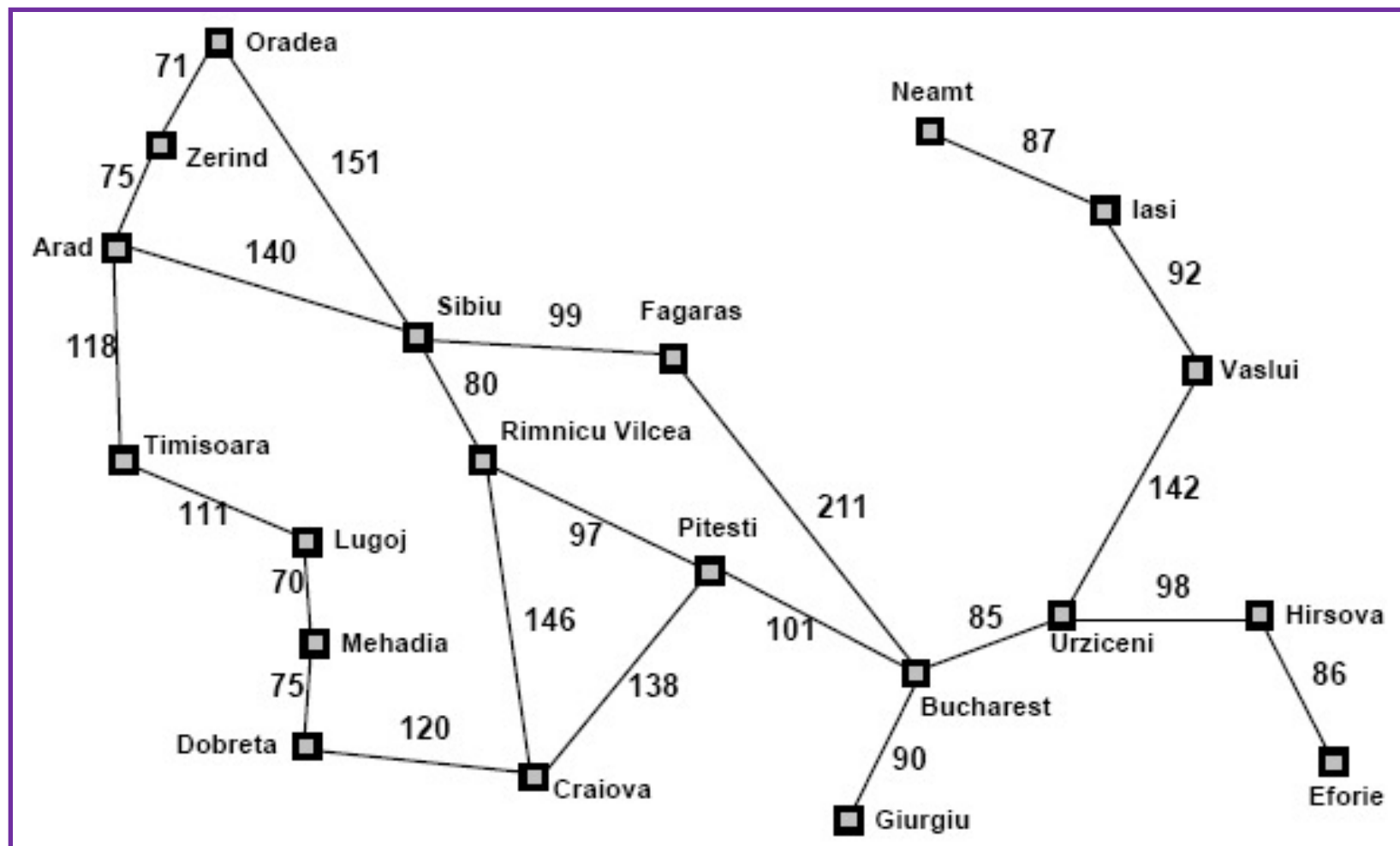
Permite saber si el estado actual es un estado meta

Búsqueda no informada

Problemas bien definidos

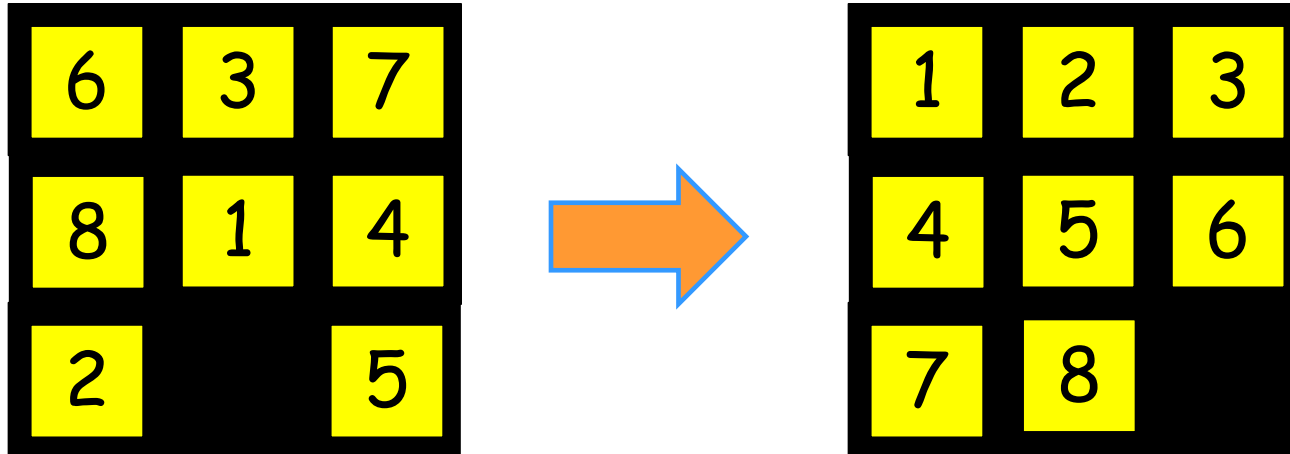
- Función costo de ruta
 - Se le asigna un costo a una ruta determinada
 - El costo es la suma de los costos de cada una de las acciones individuales a lo largo de una ruta
 - Se denota por la letra g

Búsqueda no informada



Búsqueda no informada

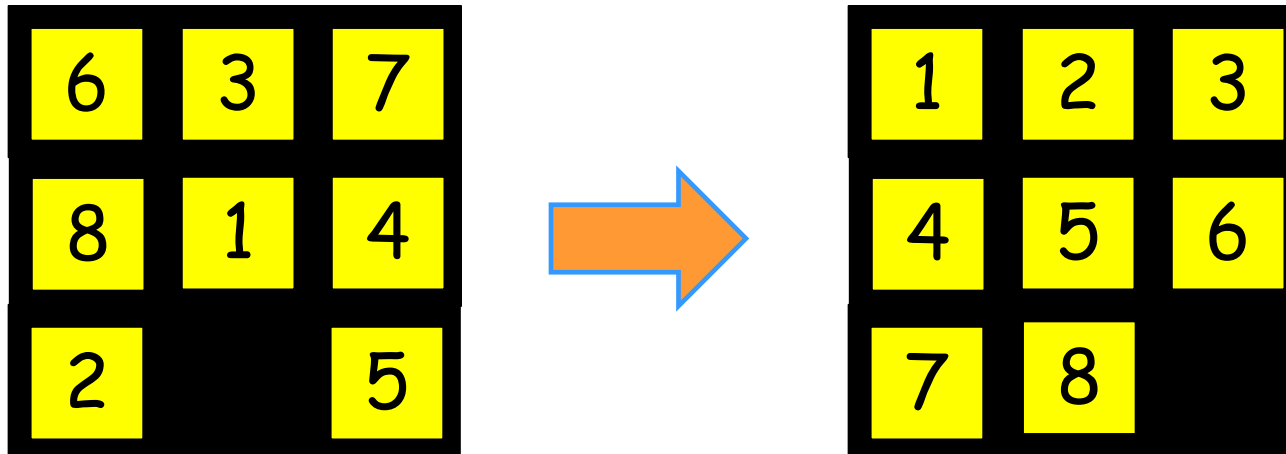
8-puzzle



- Estados
- Operadores
- Prueba de meta
- Costo de ruta

Búsqueda no informada

8-puzzle



- **Estados:** configuración de 8 piezas en el tablero

Búsqueda no informada

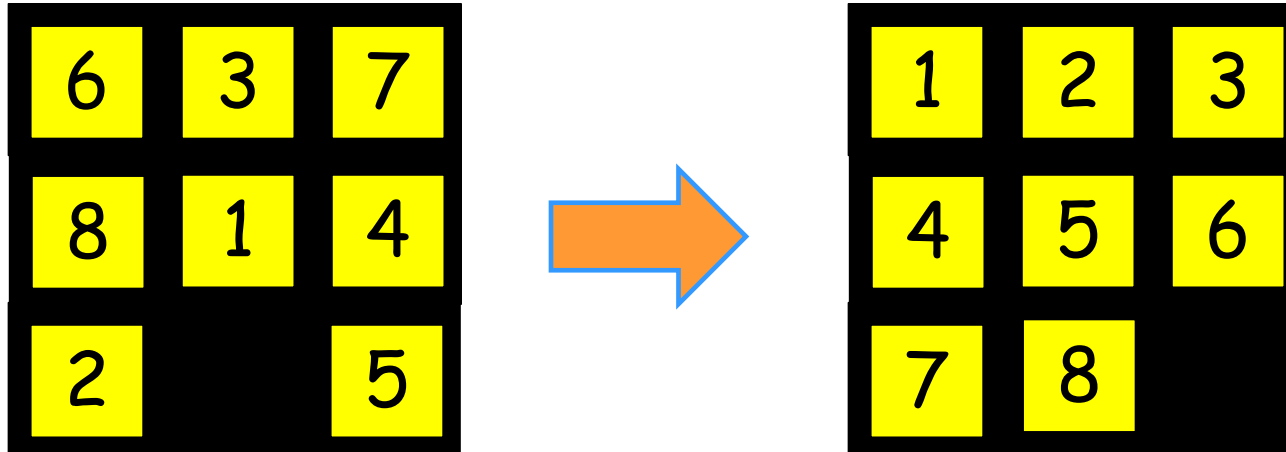
8-puzzle

```
int puzzle[][]=new int[3][3];  
puzzle[0][0]=6;  
puzzle[0][1]=3;  
puzzle[0][2]=7;  
puzzle[1][0]=8;  
puzzle[1][1]=1;  
puzzle[1][2]=4;  
puzzle[2][0]=2;  
puzzle[2][1]=0;  
puzzle[2][2]=5;
```

6	3	7
8	1	4
2		5

Búsqueda no informada

8-puzzle



- Operadores:
 - Mover espacio arriba
 - Mover espacio abajo
 - Mover espacio izquierda
 - Mover espacio derecha

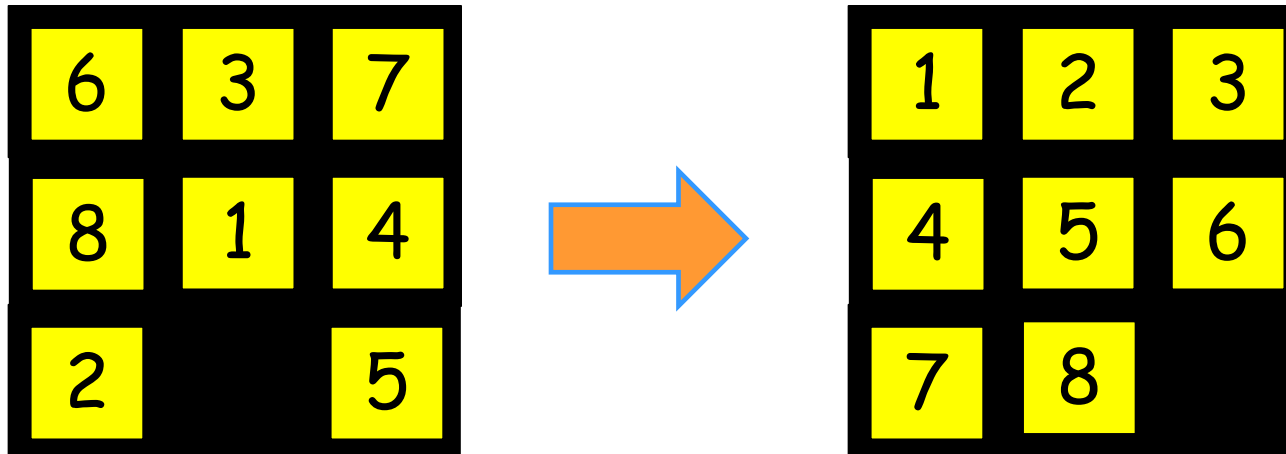
Búsqueda no informada

8-puzzle

```
public void mover(int direccion)
    if (direccion==1) //1 es izquierda, 2 es arriba ...
        for (int i=0; i<3; i++)
            for (int j=0; j<3; j++)
                if (puzzle[i][j]==0){
                    fil=i;
                    col=j;
                }
    puzzle[fil][col]=puzzle[fil][col-1];
    puzzle[fil][col-1]=0;
```

Búsqueda no informada

8-puzzle



- Prueba de meta: configuración ordenada (*ver figura*)

Búsqueda no informada

8-puzzle

```
if (puzzle[0][0]==1 && puzzle[0][1]==2 && puzzle[0][2]==3 &&  
    puzzle[1][0]==4 && puzzle[1][1]==5 && puzzle[1][2]==6 &&  
    puzzle[2][0]==7 && puzzle[2][1]==8 && puzzle[2][2]==0 ){
```

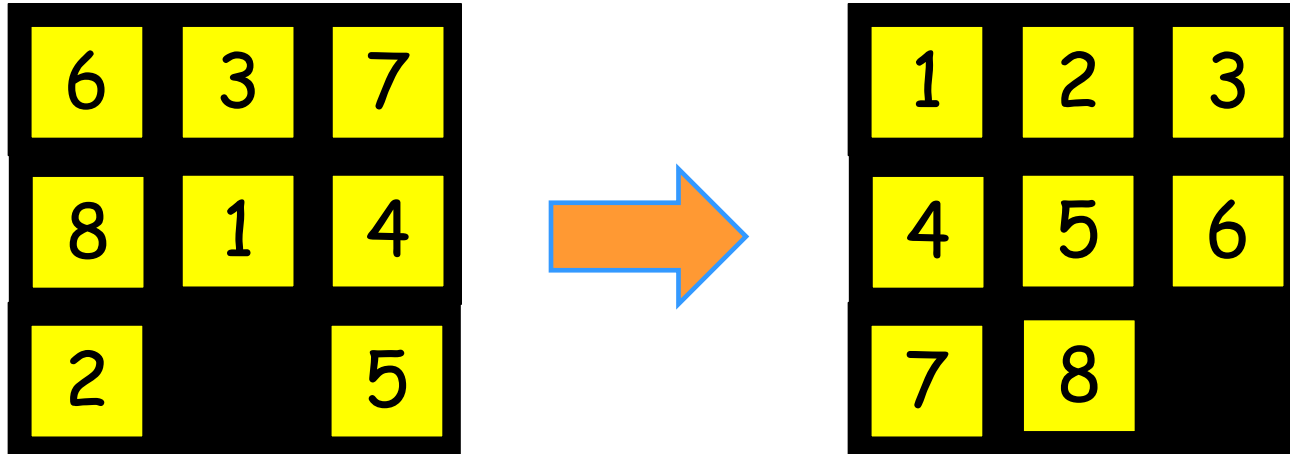
```
    System.out.println(" 😊 ");
```

```
}
```

1	2	3
4	5	6
7	8	

Búsqueda no informada

8-puzzle



- **Costo de ruta:** cada movimiento cuesta 1, esto es, el costo de una ruta corresponde a la cantidad de movimientos

Búsqueda no informada

8-puzzle

```
public int g(String[] movimientos){  
    return length(movimientos)  
}
```

Búsqueda no informada

El mundo de la aspiradora

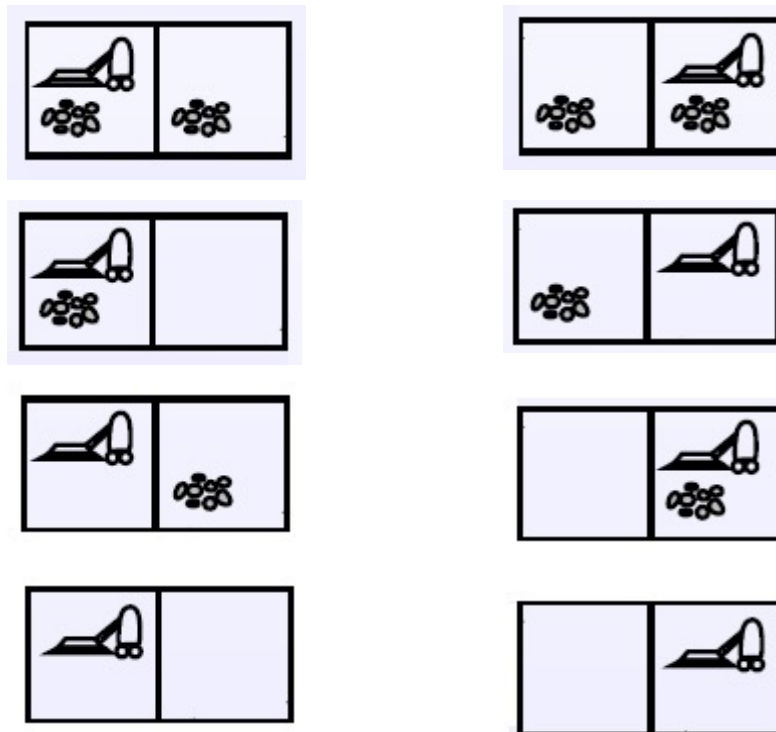


- Estados
- Operadores
- Prueba de meta
- Costo de ruta

Búsqueda no informada

El mundo de la aspiradora

- Estados: son 8 posibles estados



Búsqueda no informada

El mundo de la aspiradora

- **Operadores:**
 - Mover la aspiradora a la izquierda (I)
 - Mover la aspiradora a la derecha (D)
 - Aspirar (A)

Búsqueda no informada

El mundo de la aspiradora

- Operadores:
 - Mover la aspiradora a la izquierda (I)
 - Mover la aspiradora a la derecha (D)
 - Aspirar (A)
- Defina los métodos:
 - moverIzquierda()
 - aspirar()

Búsqueda no informada

El mundo de la aspiradora

- Prueba de meta:
 - No hay mugre en ninguno de los cuartos

Búsqueda no informada

El mundo de la aspiradora

- Costo de ruta
 - Aspirar: 3
 - Desplazarse: 1

Búsqueda no informada

Árboles de búsqueda

- Permite facilitar la comprensión de las **exploraciones**
 - El estado inicial es el **nodo raíz**
 - Se **expande** el nodo raíz al aplicar los operadores sobre el nodo
 - En cada paso, el algoritmo de búsqueda escoge un nodo "hoja" y lo expande

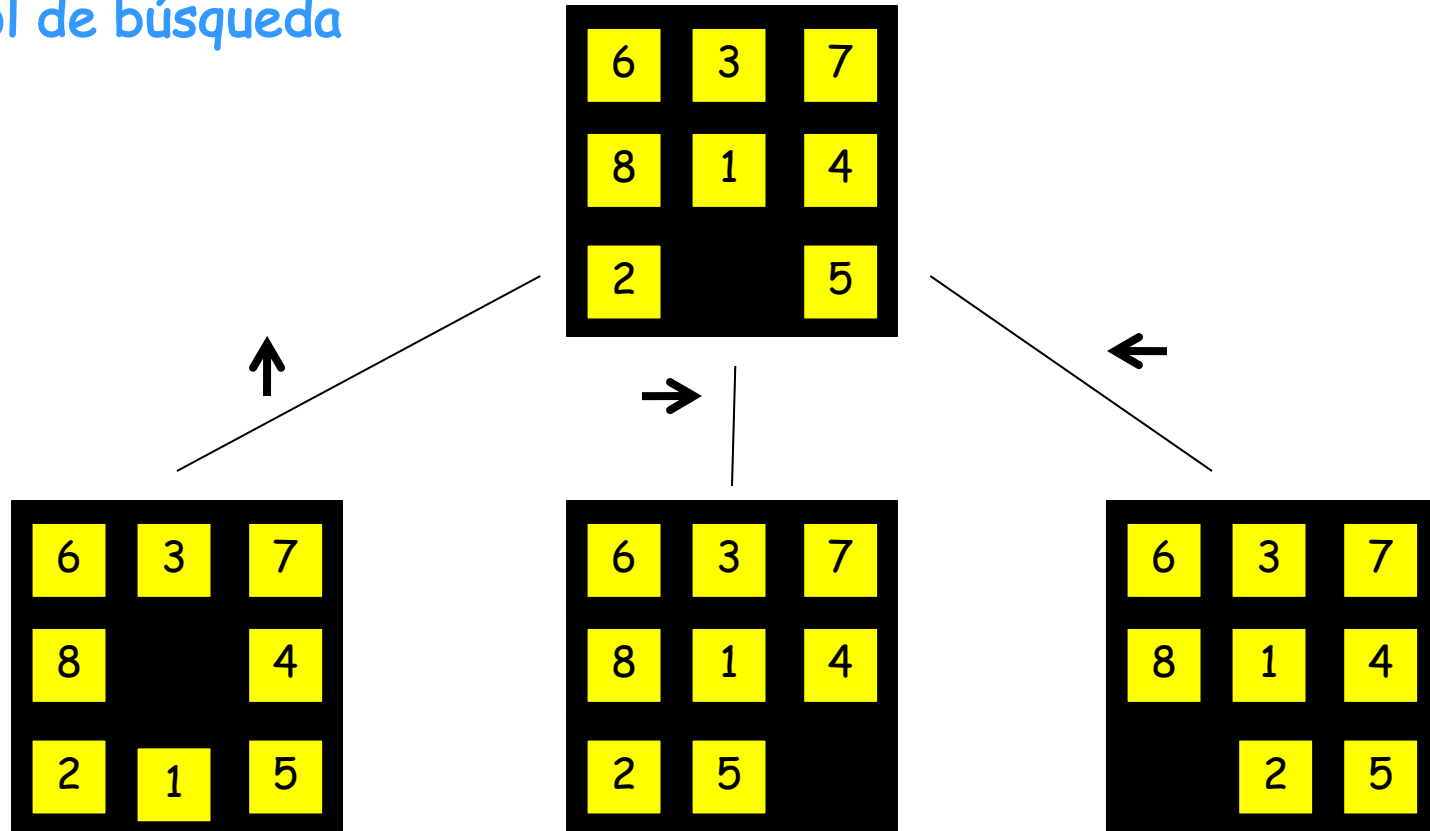
Búsqueda no informada

Árbol de búsqueda

6	3	7
8	1	4
2		5

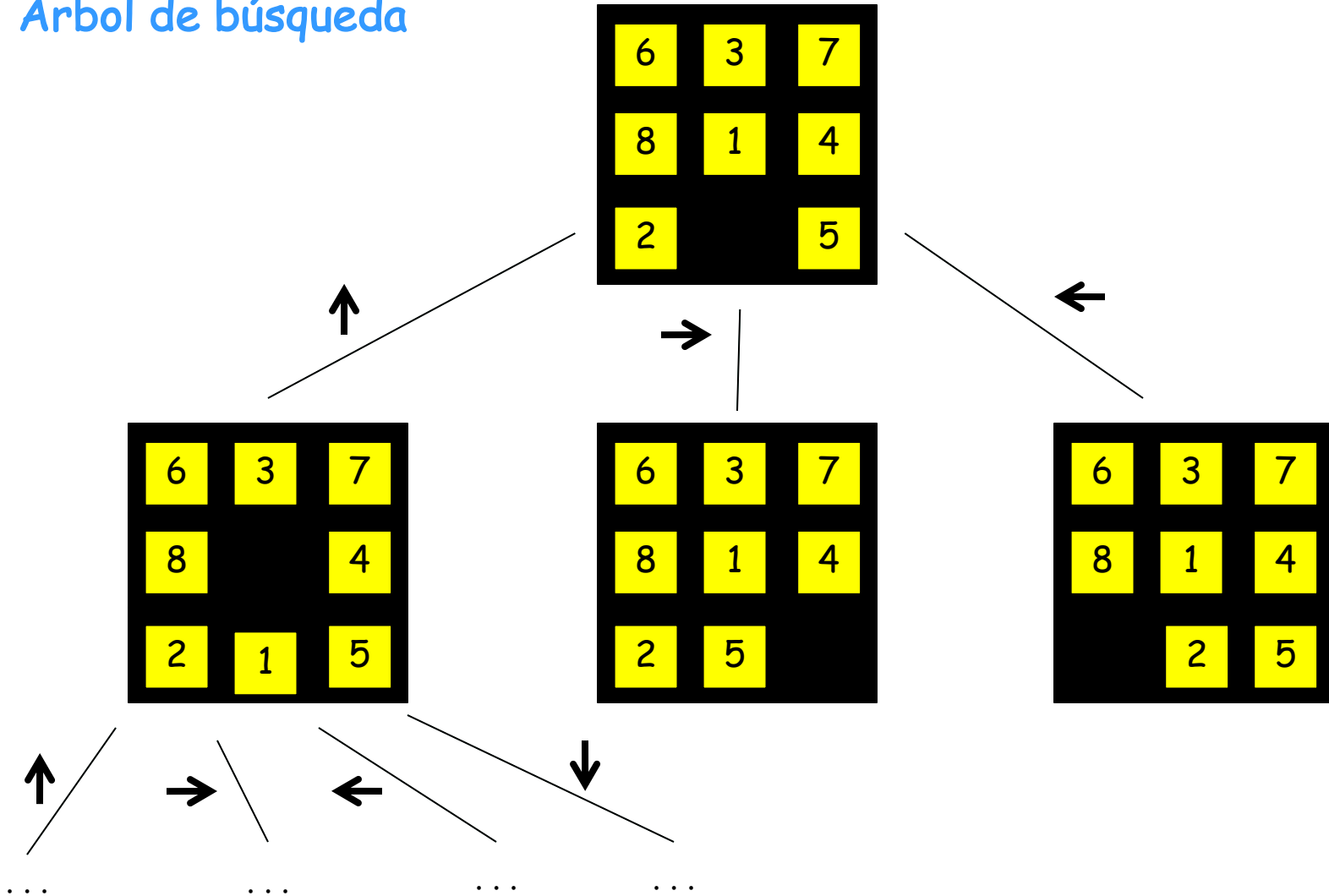
Búsqueda no informada

Árbol de búsqueda



Búsqueda no informada

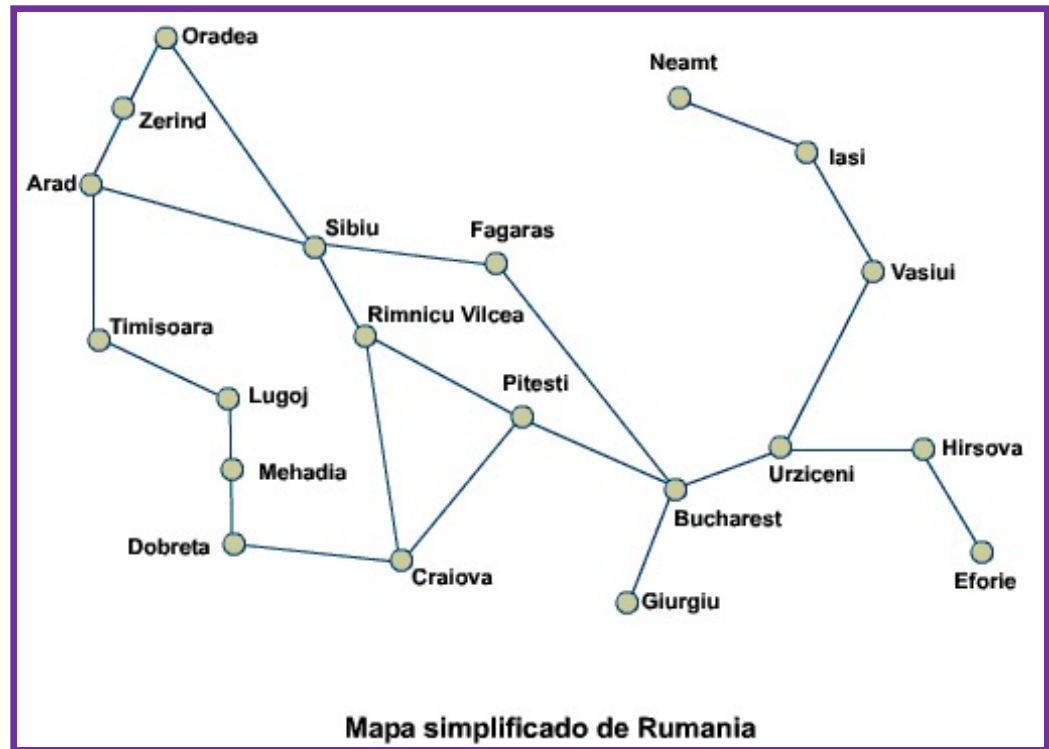
Árbol de búsqueda



Búsqueda no informada

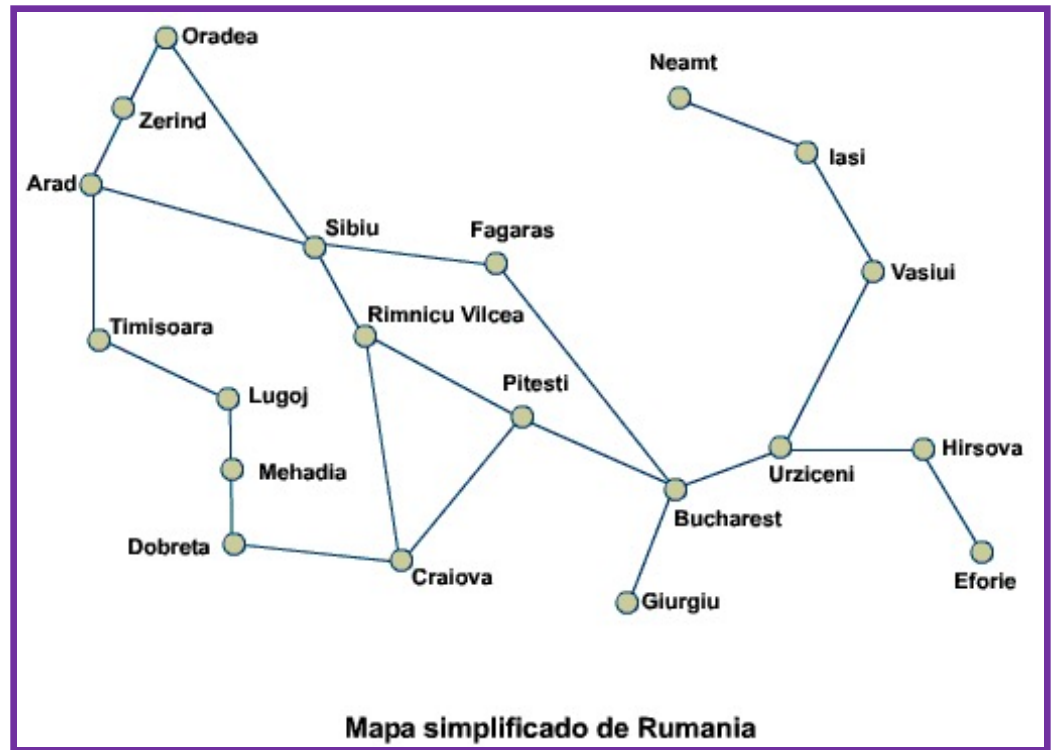
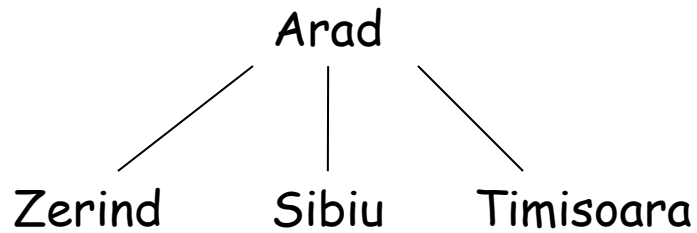
Árbol de búsqueda

Arad



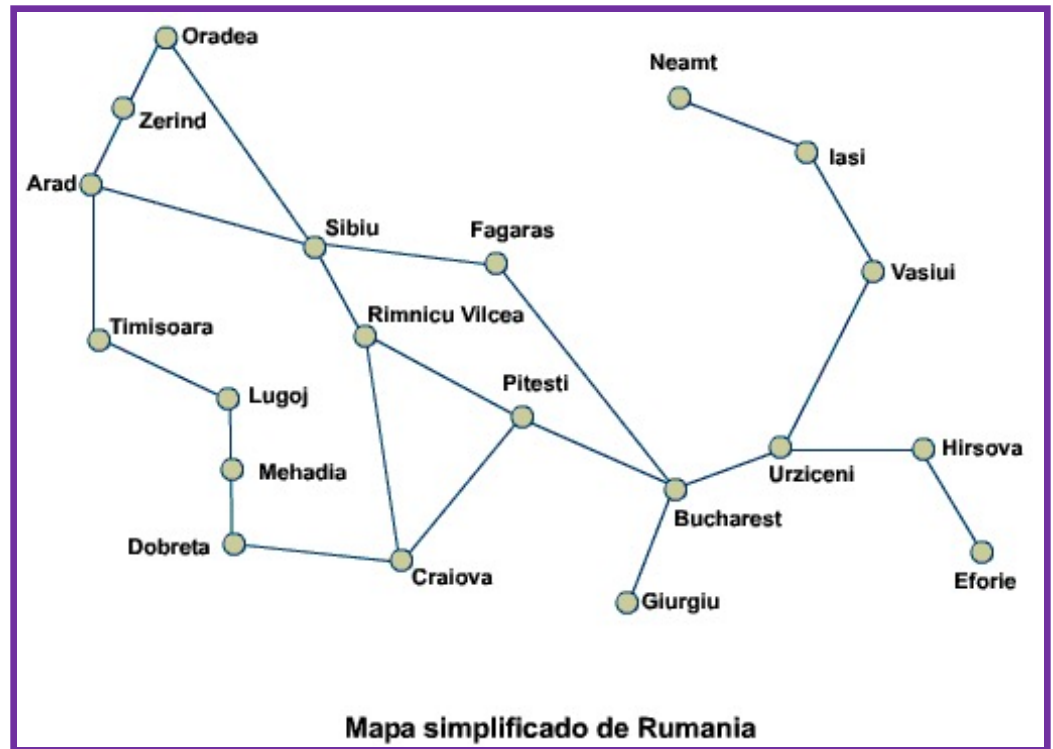
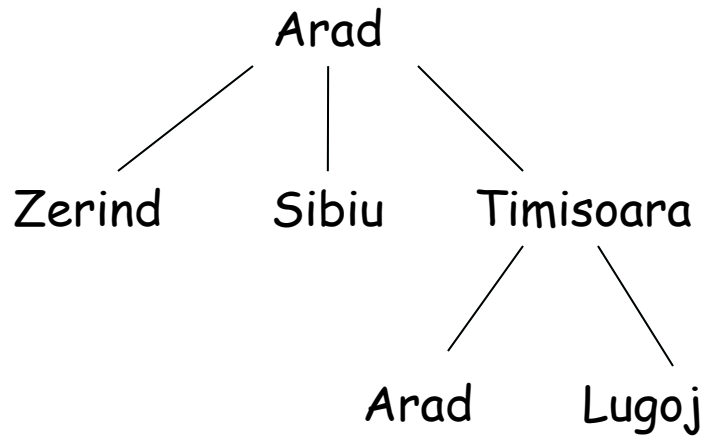
Búsqueda no informada

Árbol de búsqueda



Búsqueda no informada

Árbol de búsqueda



Búsqueda no informada

Árboles de búsqueda

Cada nodo del árbol guarda la siguiente información:

- El estado del problema
- Una referencia al nodo padre
- El operador que se aplicó para generar el nodo
- Profundidad del nodo
- El costo de la ruta desde la raíz hasta el nodo

Búsqueda no informada

Árboles de búsqueda

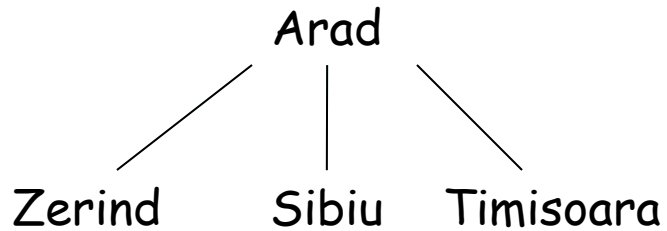
Cada nodo del árbol guarda la siguiente información:

- El estado del problema
- Una referencia al nodo padre
- El operador que se aplicó para generar el nodo
- Profundidad del nodo
- El costo de la ruta desde la raíz hasta el nodo

Métodos de un nodo:

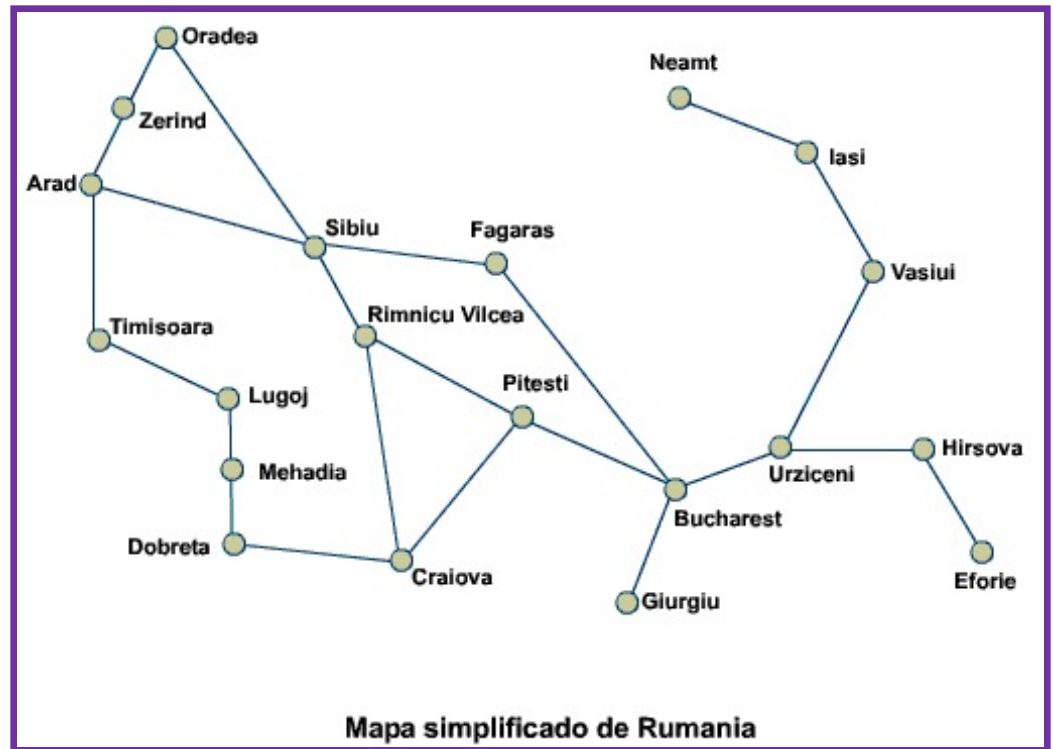
- `aplicarOperador()`
- `esMeta()`

Búsqueda no informada

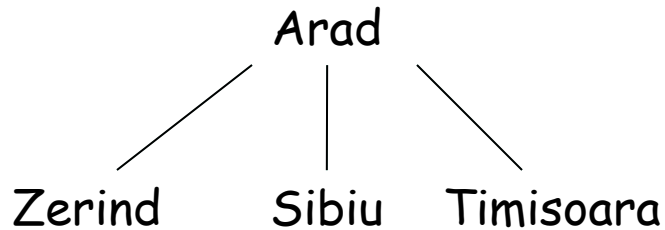


Métodos de búsqueda

- Búsqueda no informada
- Búsqueda informada

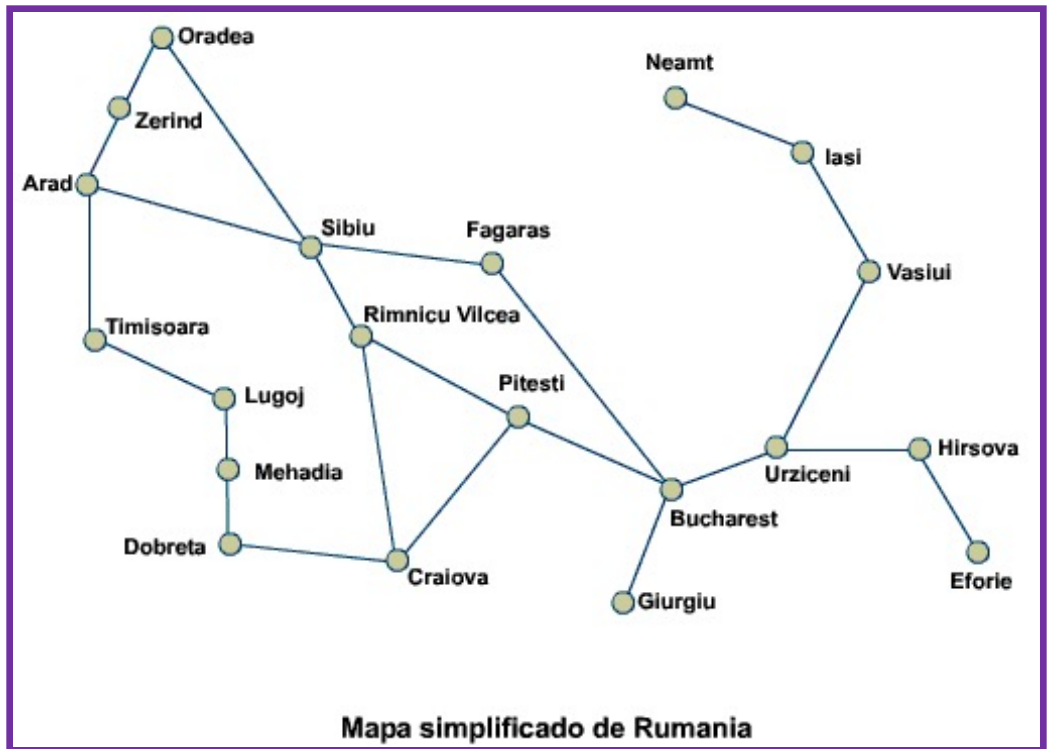


Búsqueda no informada



Búsqueda no informada

- No hay caminos prioritarios
- La única diferencia entre las técnicas es el orden en que se expanden los nodos



Búsqueda no informada

Búsqueda no informada:

- Preferente por amplitud
- De costo uniforme
- Preferente por profundidad
- Limitada por profundidad
- Por profundización iterativa

Búsqueda no informada

Búsqueda no informada:

- Preferente por amplitud
- De costo uniforme
- Preferente por profundidad
- Limitada por profundidad
- Por profundización iterativa

Búsqueda no informada

Búsqueda preferente por amplitud

- Expandir el nodo raíz, esto genera nodos de profundidad 1
- Expandir, de izquierda a derecha, cada nodo de profundidad 1
- Continuar expandiendo, de izquierda a derecha, cada uno de los nodos en la profundidad d y luego los de profundidad $d+1$

Búsqueda no informada

Búsqueda preferente por amplitud

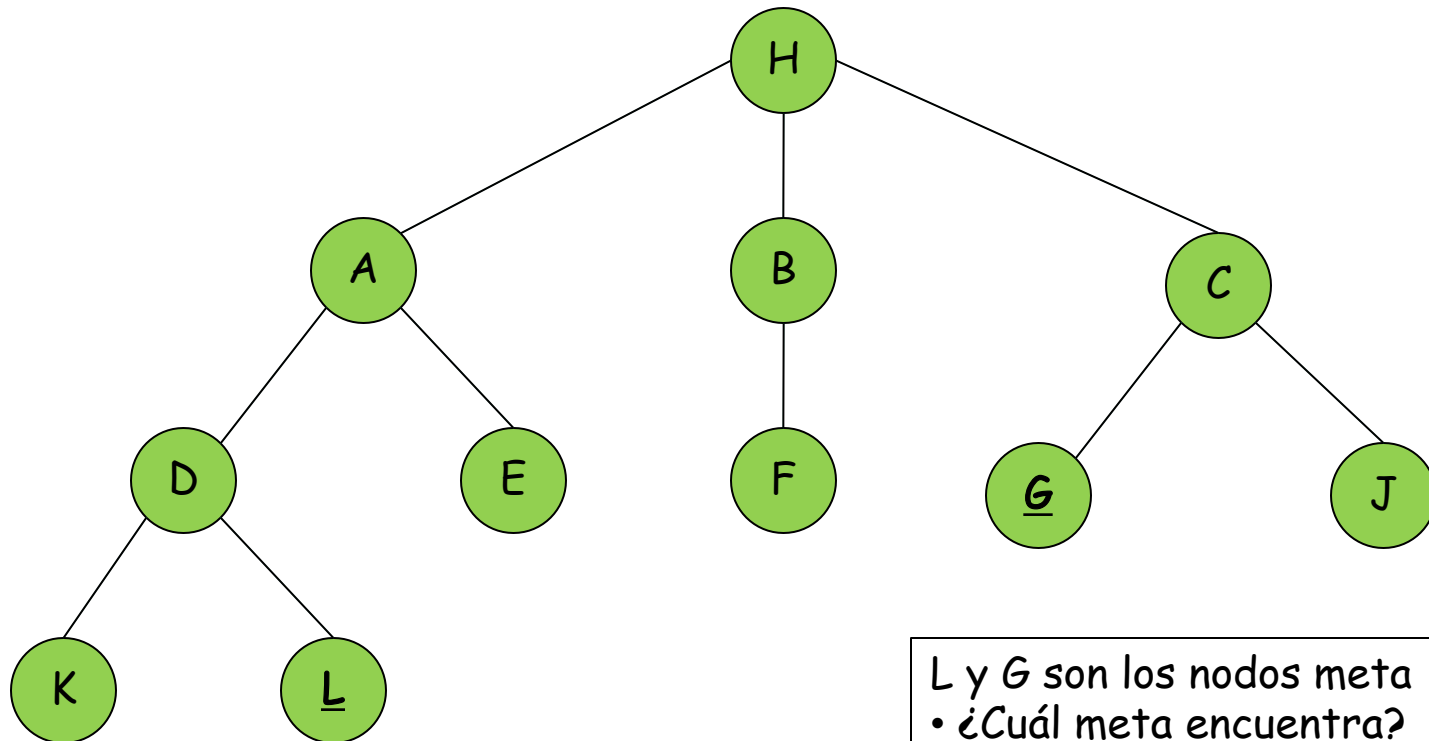
- Expandir el nodo raíz, esto genera nodos de profundidad 1
- Expandir, de izquierda a derecha, cada nodo de profundidad 1
- Continuar expandiendo, de izquierda a derecha, cada uno de los nodos en la profundidad d y luego los de profundidad $d+1$

** Expandir (verificar si es meta + crear hijos)*

** Al expandir un nodo meta no se crean sus hijos*

Búsqueda no informada

Búsqueda preferente por amplitud

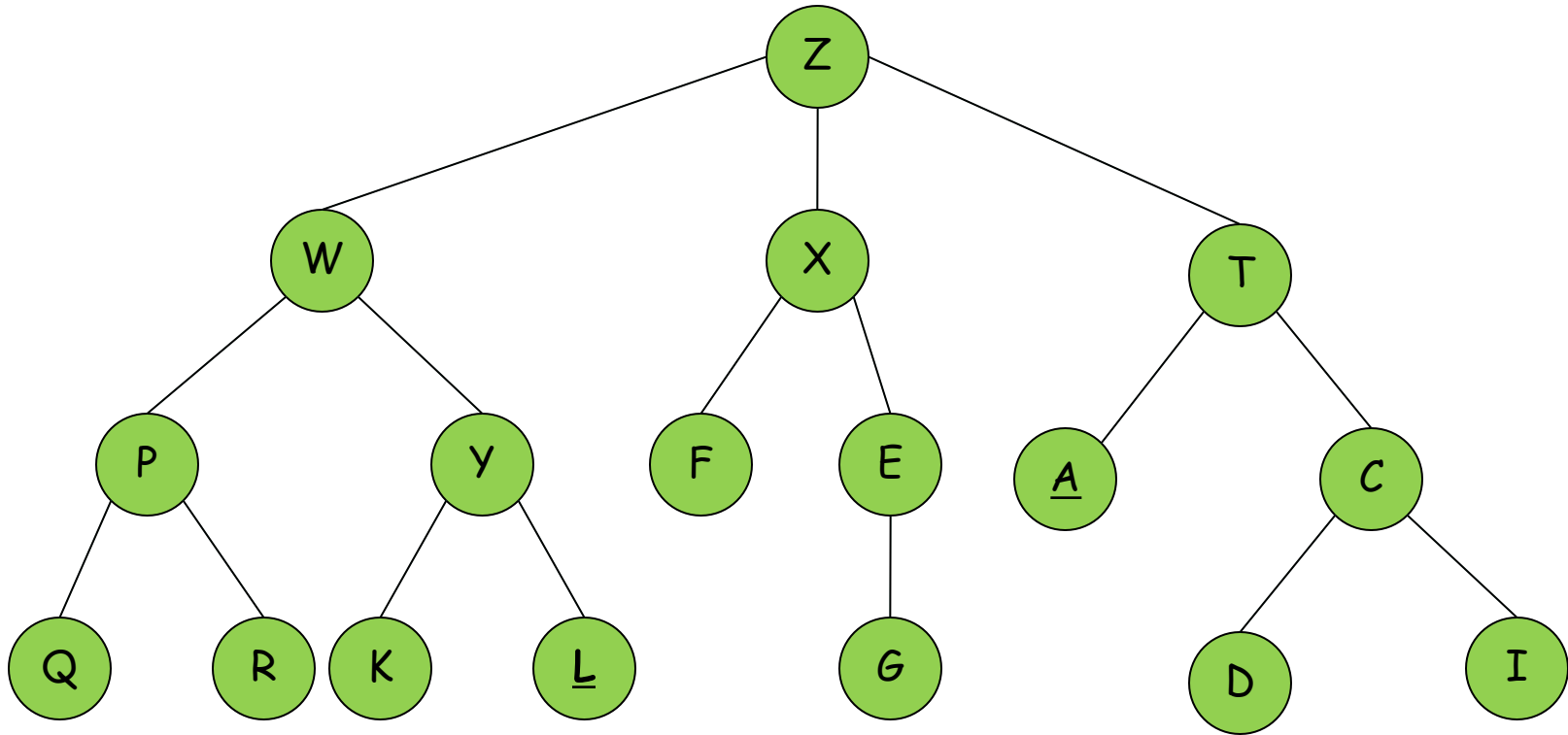


L y G son los nodos meta

- ¿Cuál meta encuentra?
- Liste los nodos creados
- Liste los nodos expandidos

Búsqueda no informada

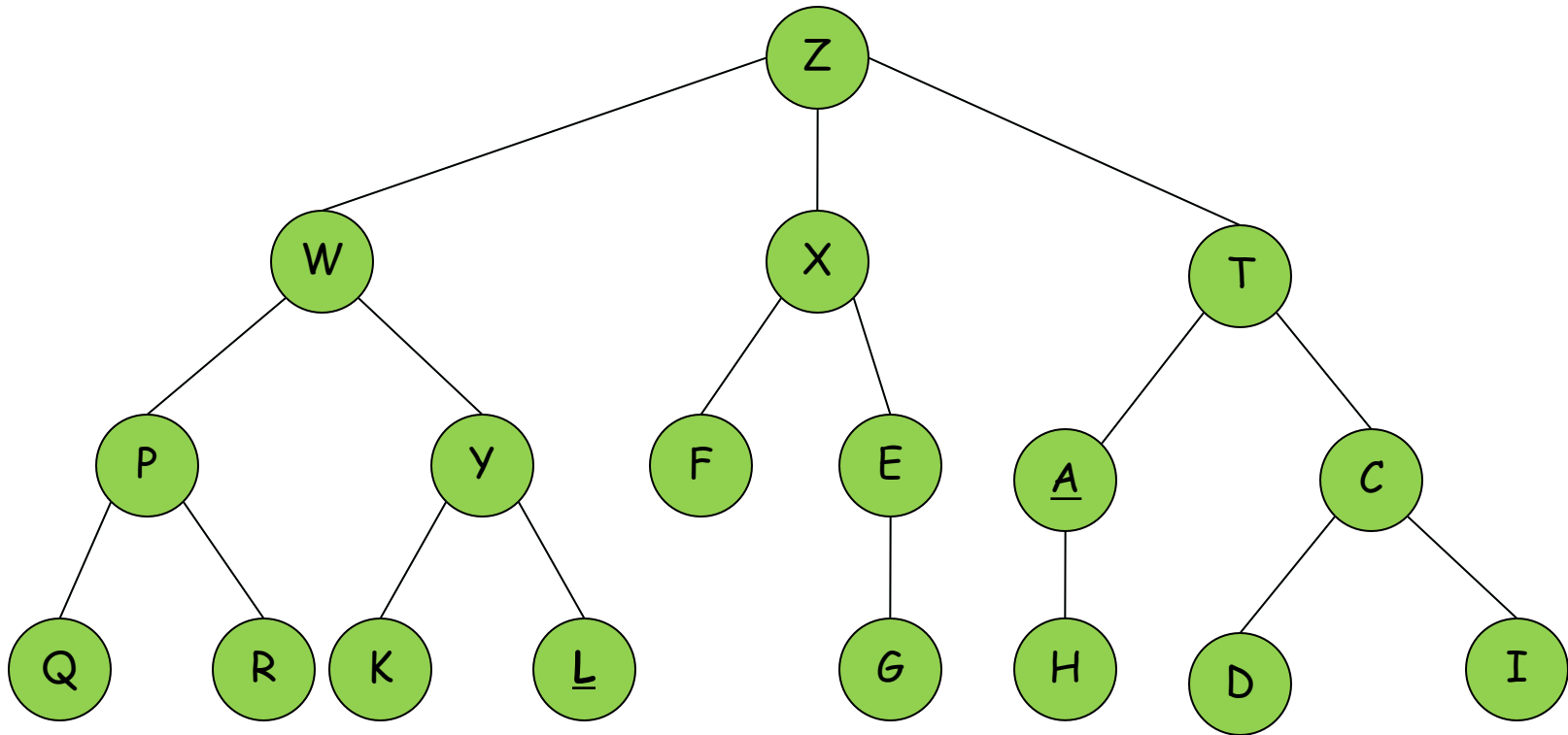
Búsqueda preferente por amplitud



- L y A son los nodos meta
- ¿Cuál meta encuentra?
 - Liste los nodos creados
 - Liste los nodos expandidos

Búsqueda no informada

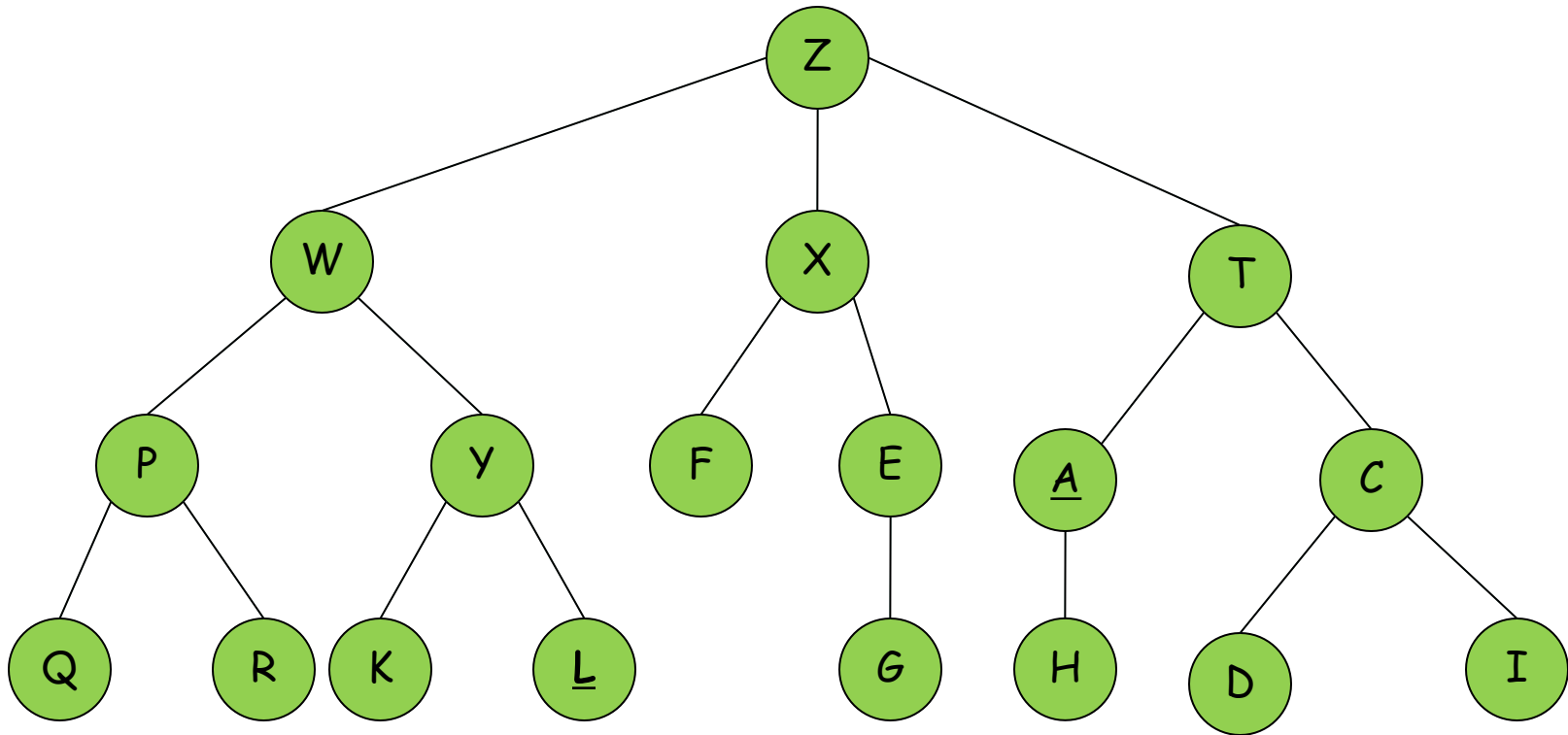
Búsqueda preferente por amplitud



- L y A son los nodos meta
- ¿Cuál meta encuentra?
 - Liste los nodos creados
 - Liste los nodos expandidos

Búsqueda no informada

Búsqueda preferente por amplitud

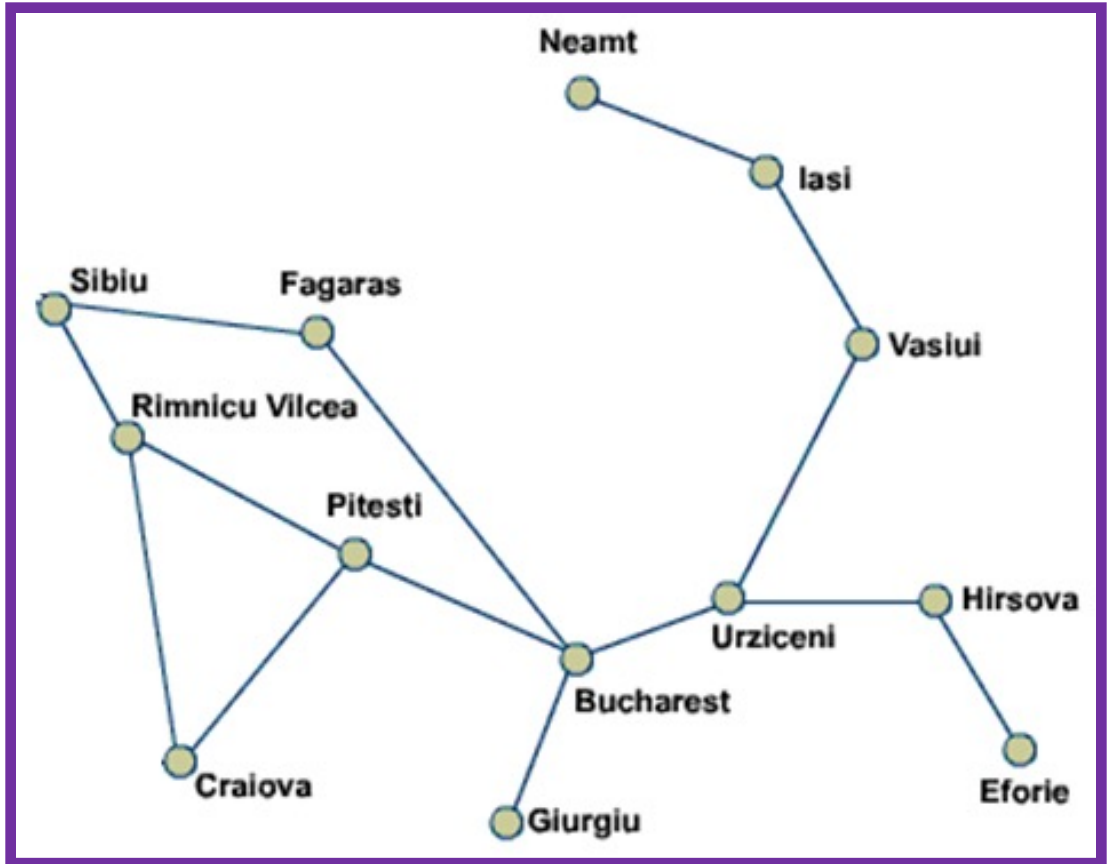


No se **CREAN** los hijos de la meta expandida

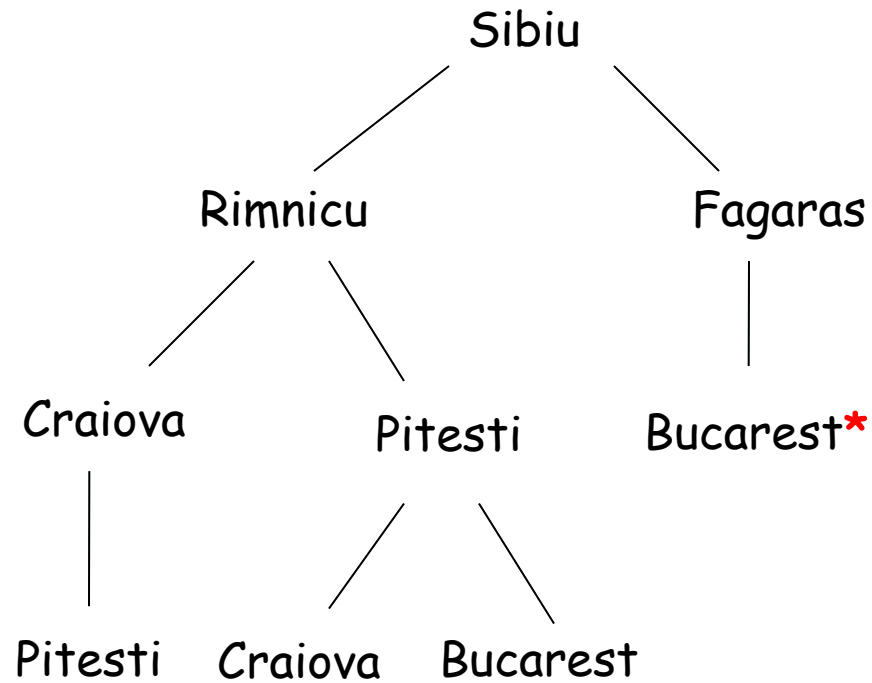
Búsqueda no informada

Aplicar búsqueda
preferente por amplitud
para ir de Sibiu a
Bucarest

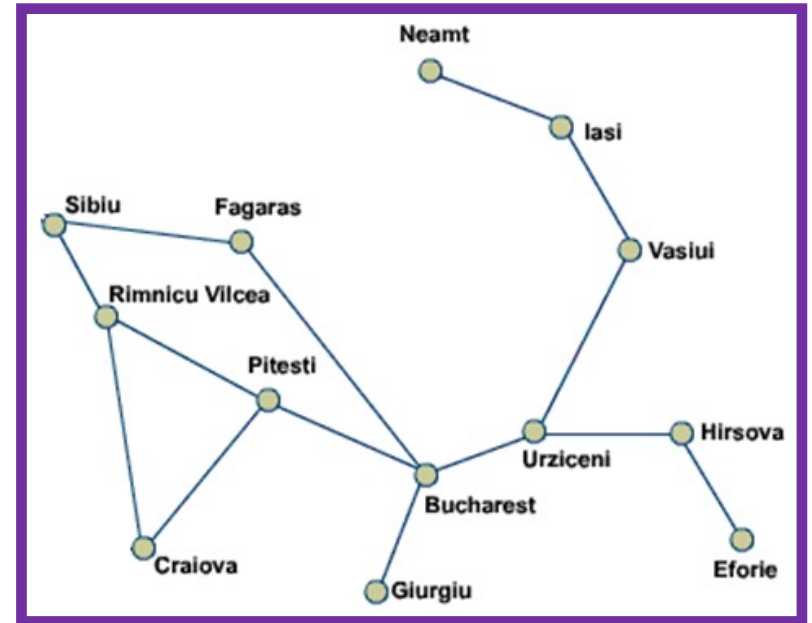
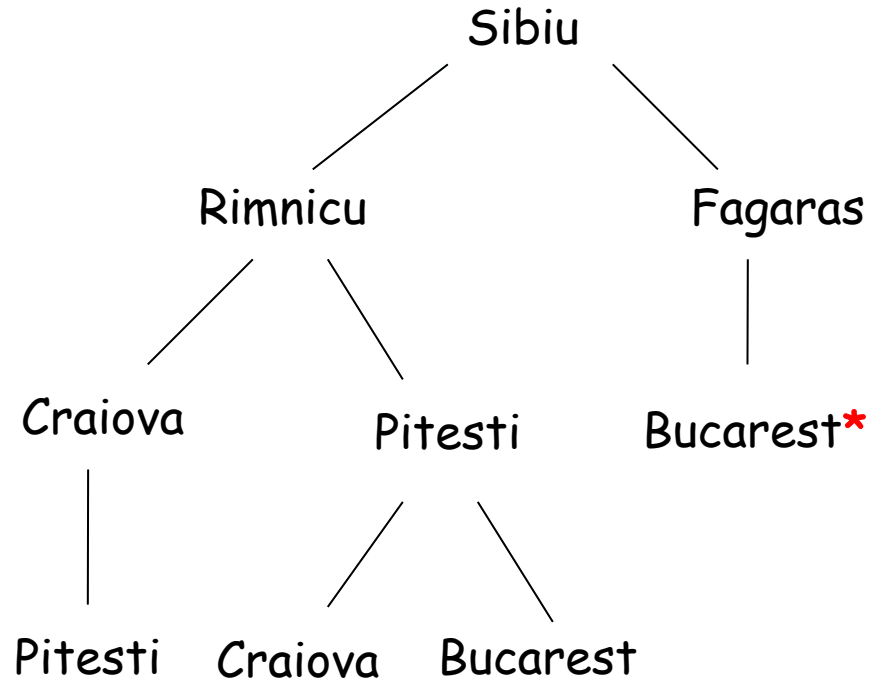
Evite devolverse



Búsqueda no informada



Búsqueda no informada



Búsqueda no informada

Aplicar búsqueda preferente por amplitud

Agente Minero. Indique el camino que sigue el agente.

Evite devolverse

 Punto de inicio

 Lingote

	0	1	2	3	4
0					
1					
2					
3					
4					

Búsqueda no informada

Aplicar búsqueda preferente por amplitud

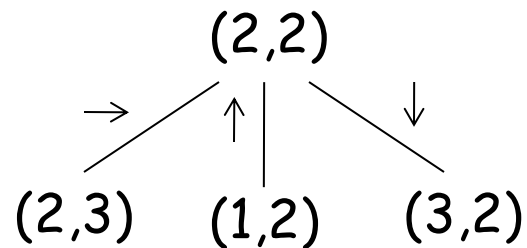
Agente Minero. Indique el camino que sigue el agente.

Evite devolverse

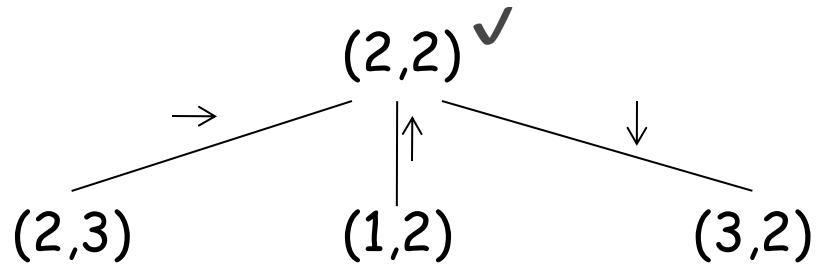
 Punto de inicio

 Lingote

	0	1	2	3	4
0					
1					
2					
3					
4					

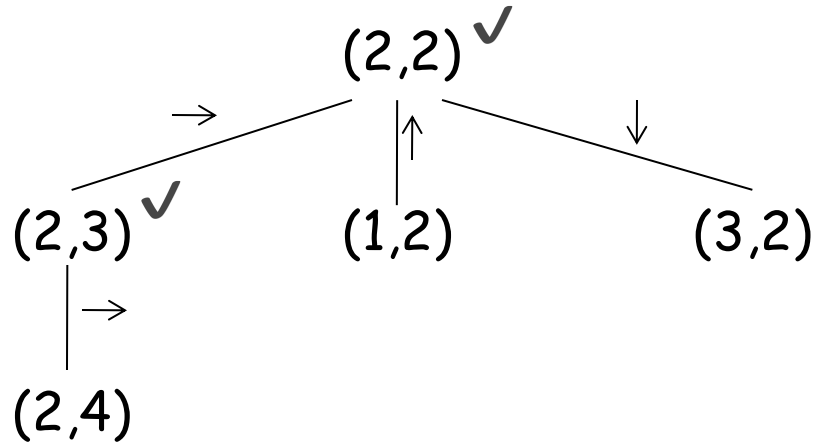


Búsqueda no informada



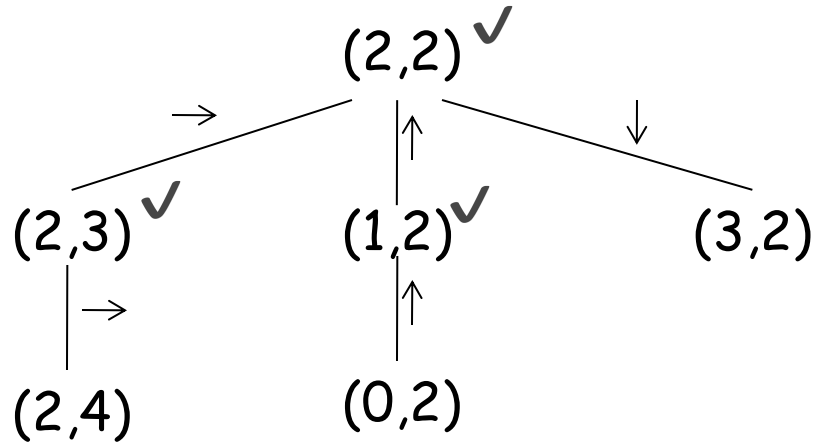
	0	1	2	3	4
0					
1					
2					
3					
4					

Búsqueda no informada



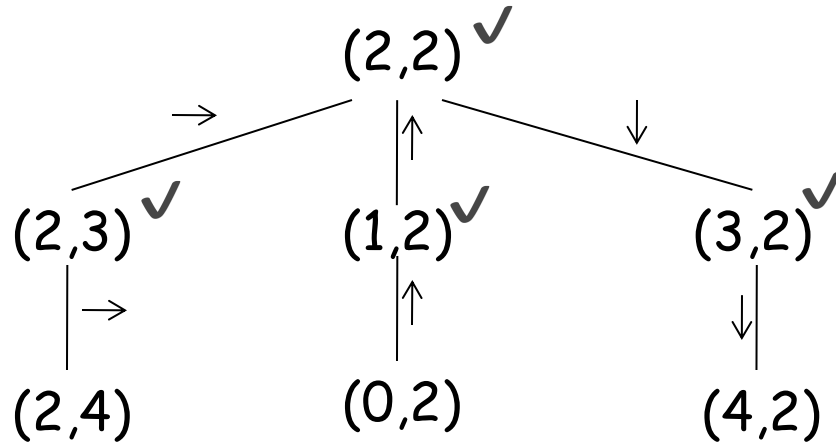
	0	1	2	3	4
0					
1					
2					
3					
4					

Búsqueda no informada



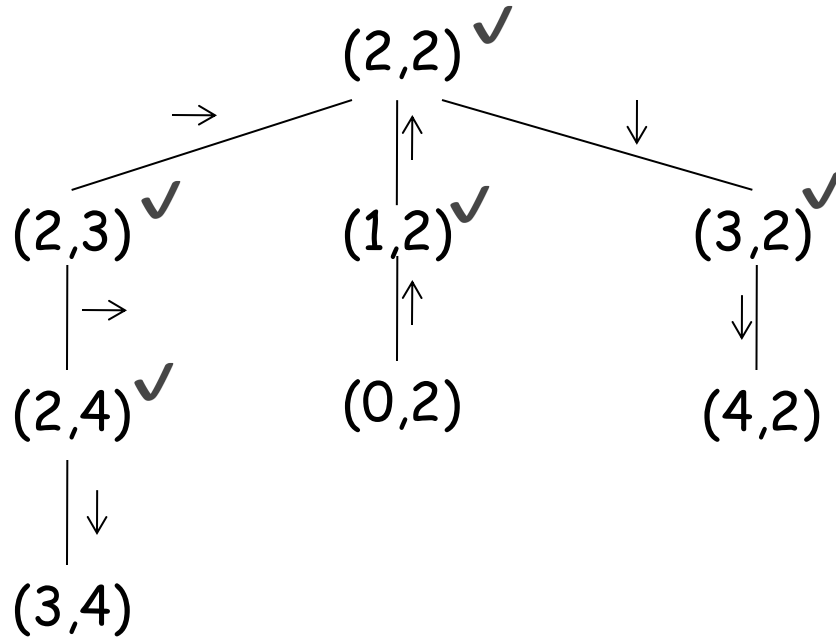
	0	1	2	3	4
0					
1					
2					
3					
4					

Búsqueda no informada



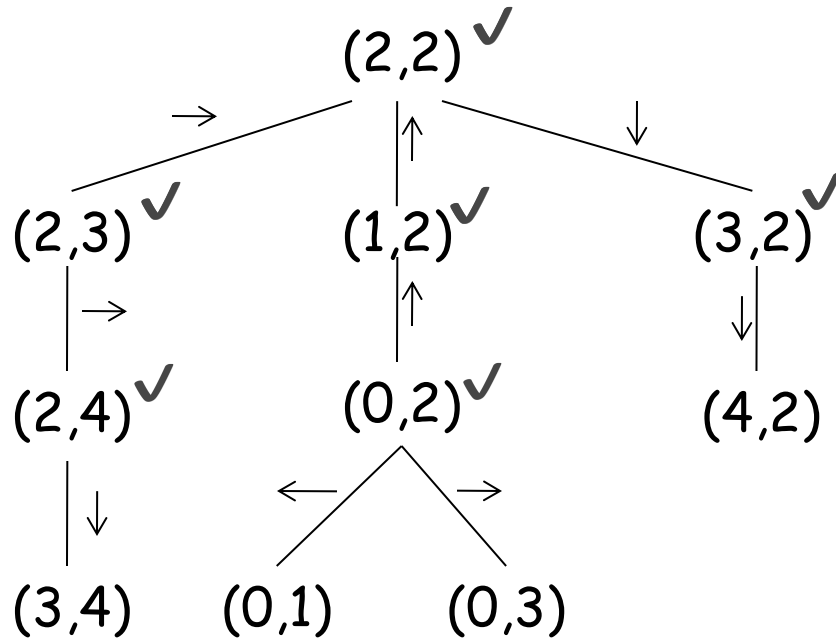
	0	1	2	3	4
0					
1					
2					
3					
4					

Búsqueda no informada



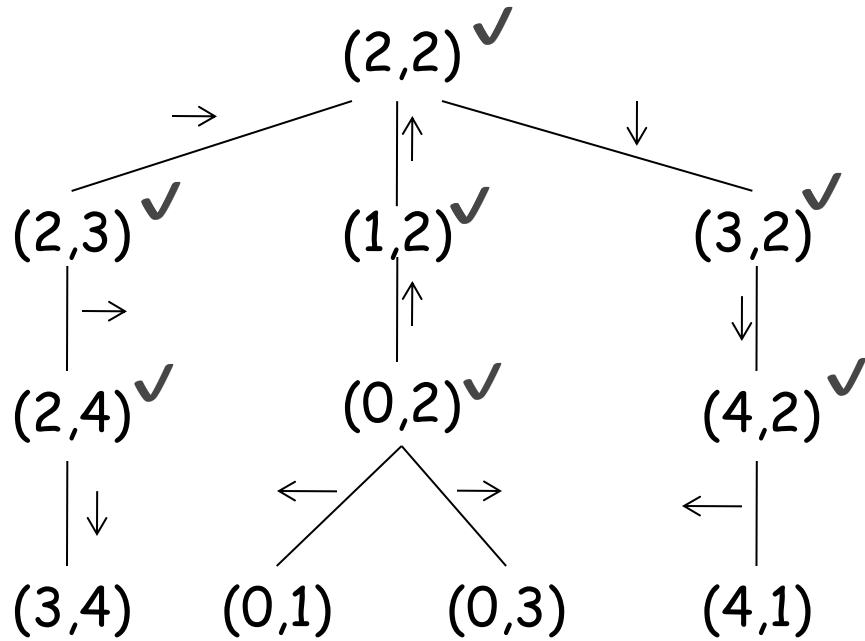
	0	1	2	3	4
0					
1					
2					
3					
4					

Búsqueda no informada



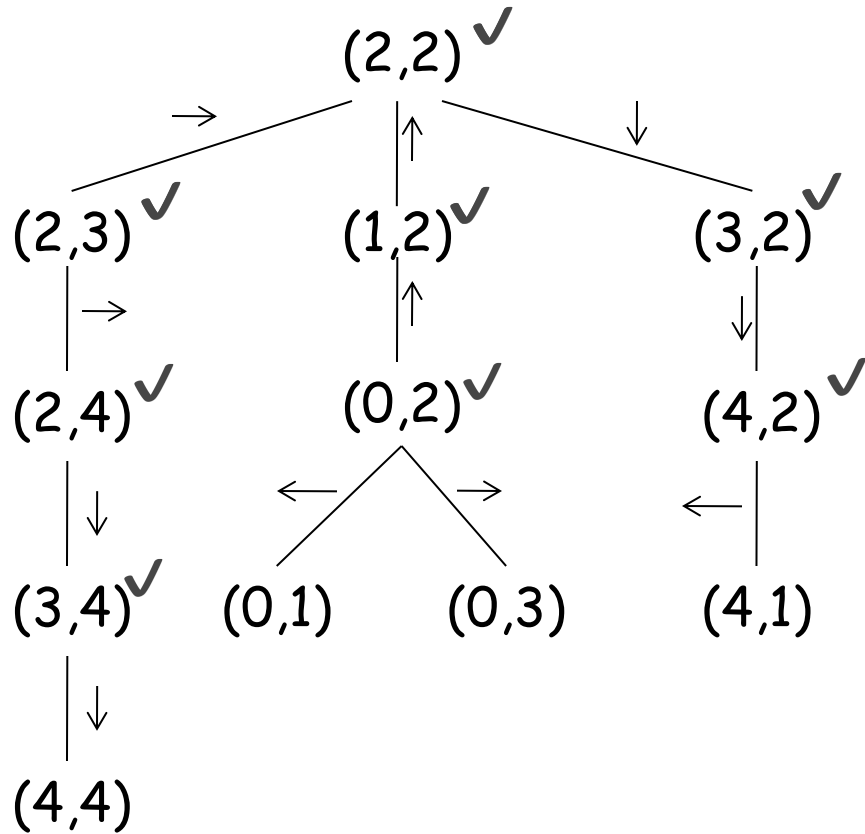
	0	1	2	3	4
0					
1					
2					
3					
4					

Búsqueda no informada



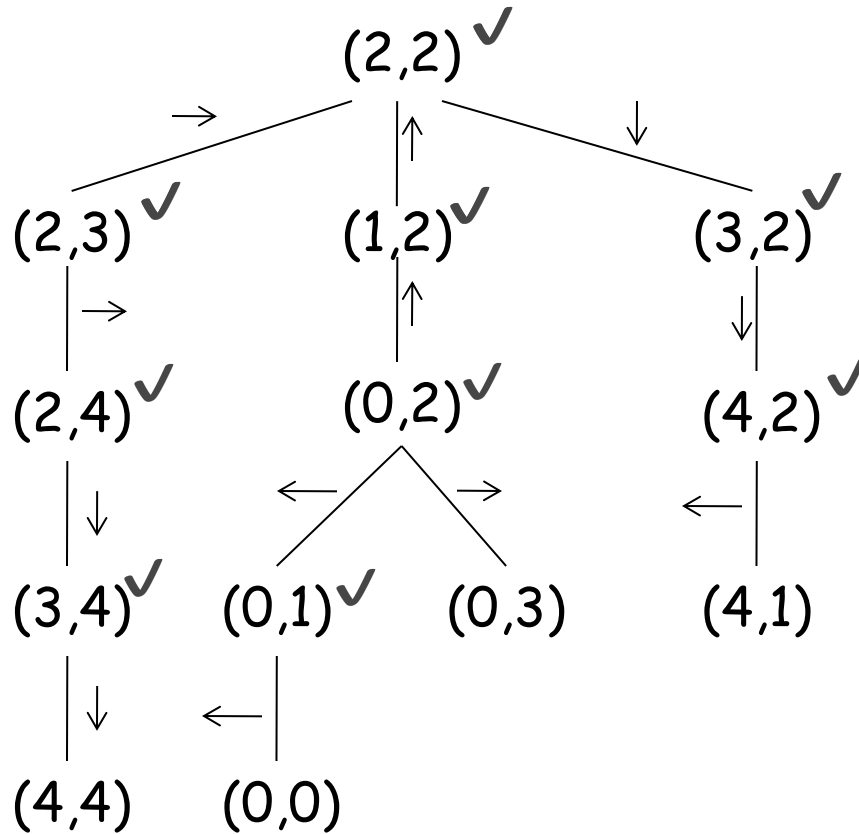
	0	1	2	3	4
0					
1					
2					
3					
4					

Búsqueda no informada



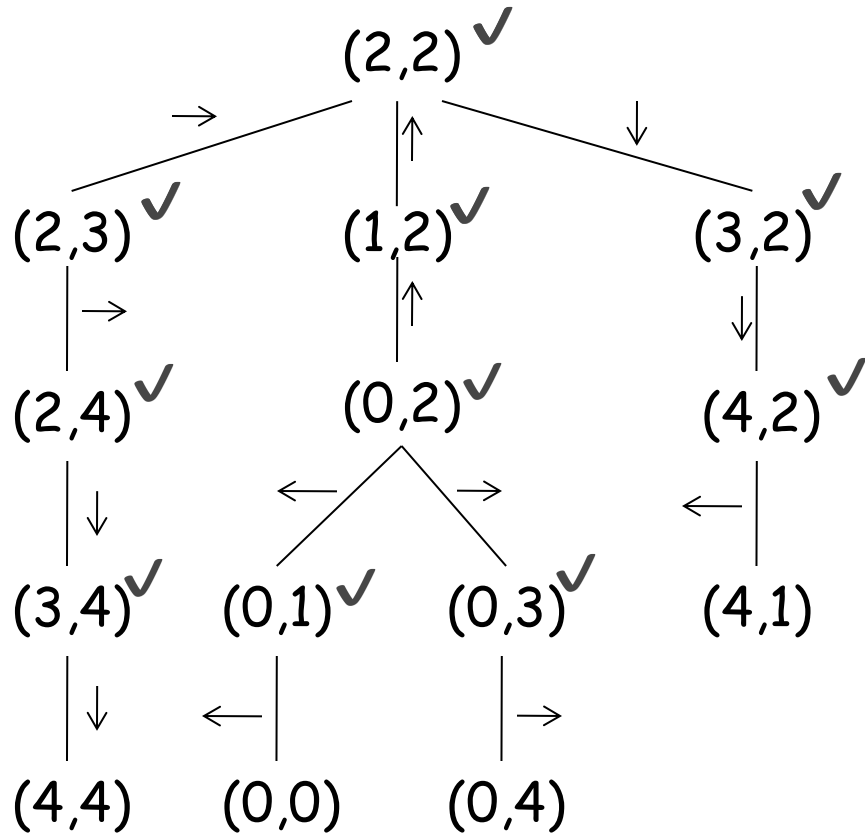
	0	1	2	3	4
0					
1					
2					
3					
4					

Búsqueda no informada



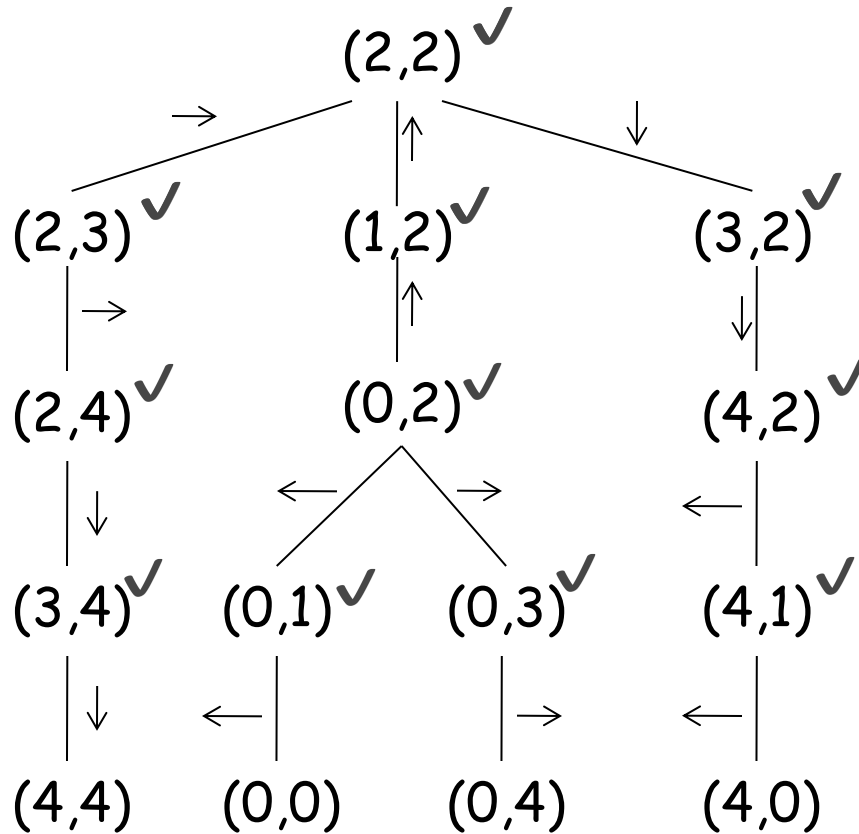
	0	1	2	3	4
0					
1					
2					
3					
4					

Búsqueda no informada



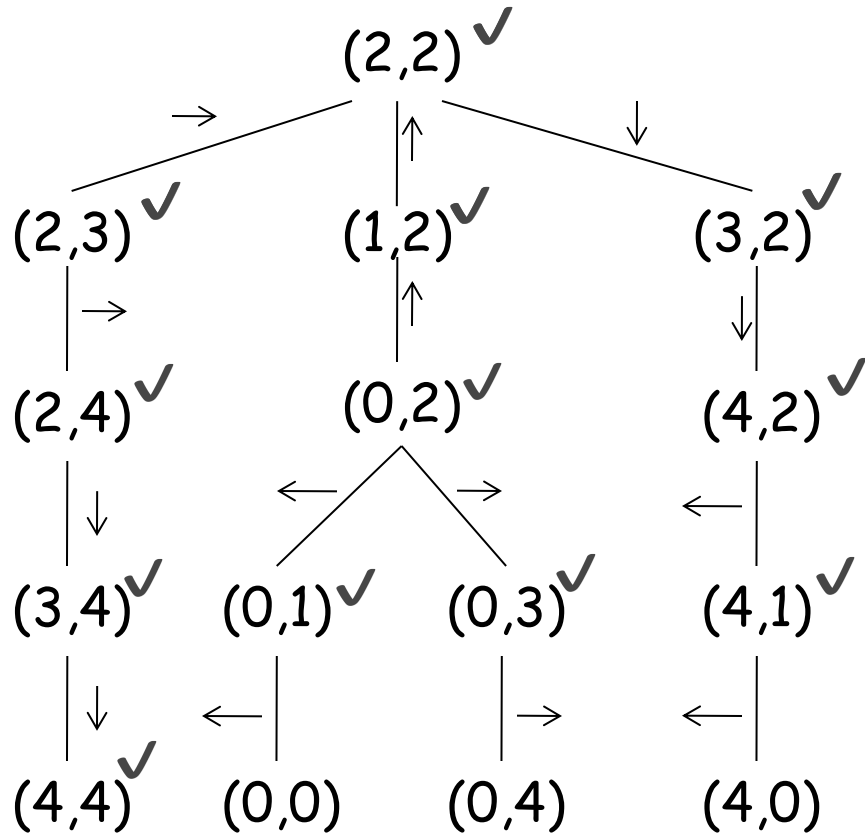
	0	1	2	3	4
0					
1					
2					
3					
4					

Búsqueda no informada



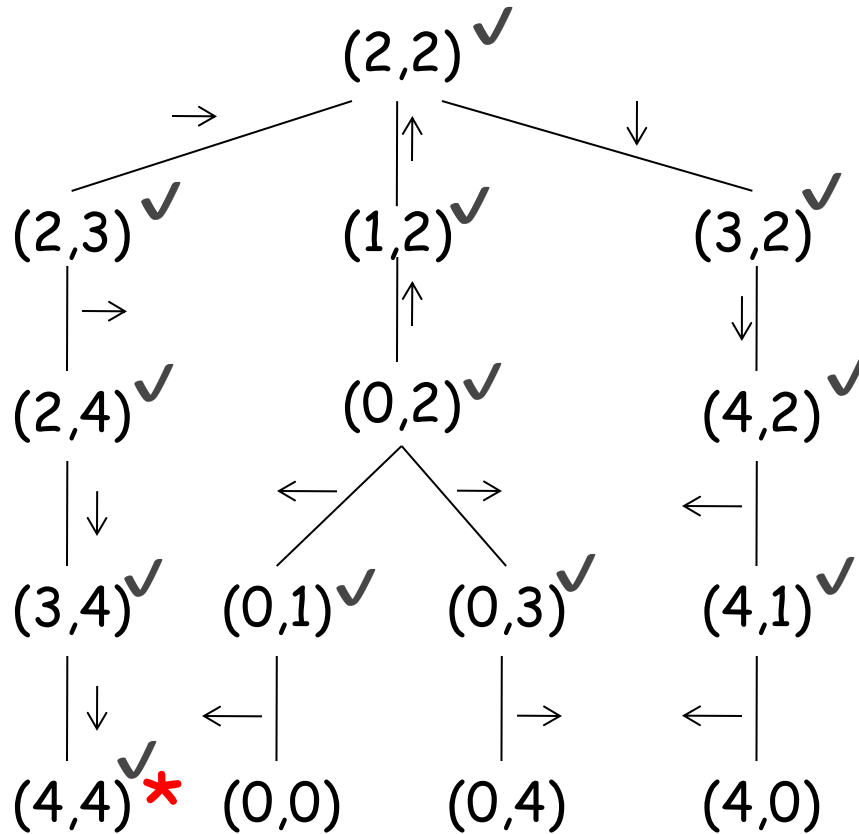
	0	1	2	3	4
0					
1					
2					
3					
4					

Búsqueda no informada



	0	1	2	3	4
0					
1					
2					
3					
4					

Búsqueda no informada

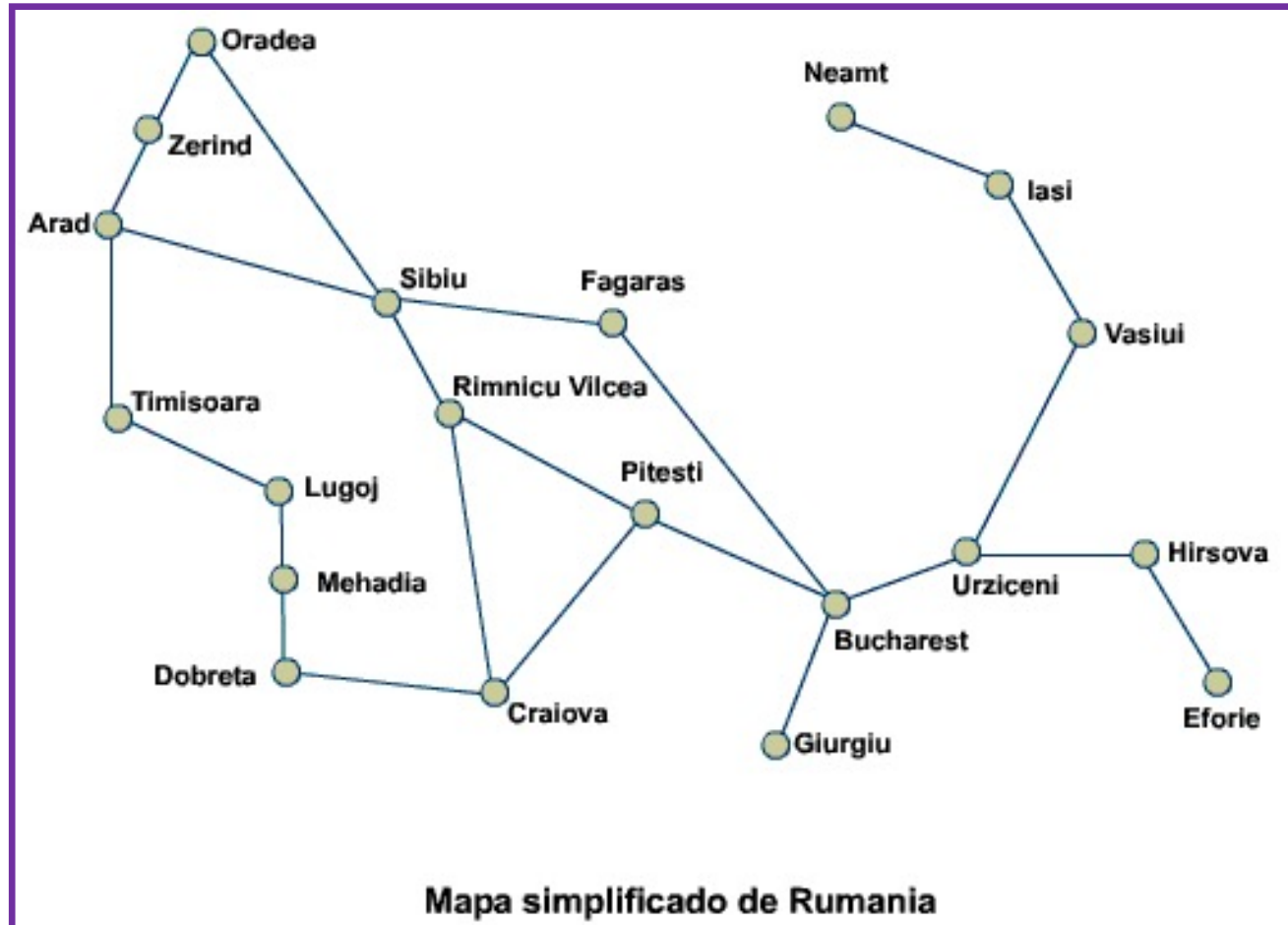
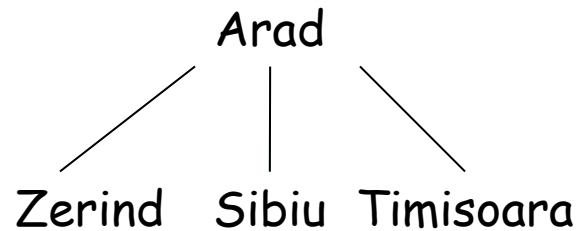


	0	1	2	3	4
0					
1					
2					
3					
4					

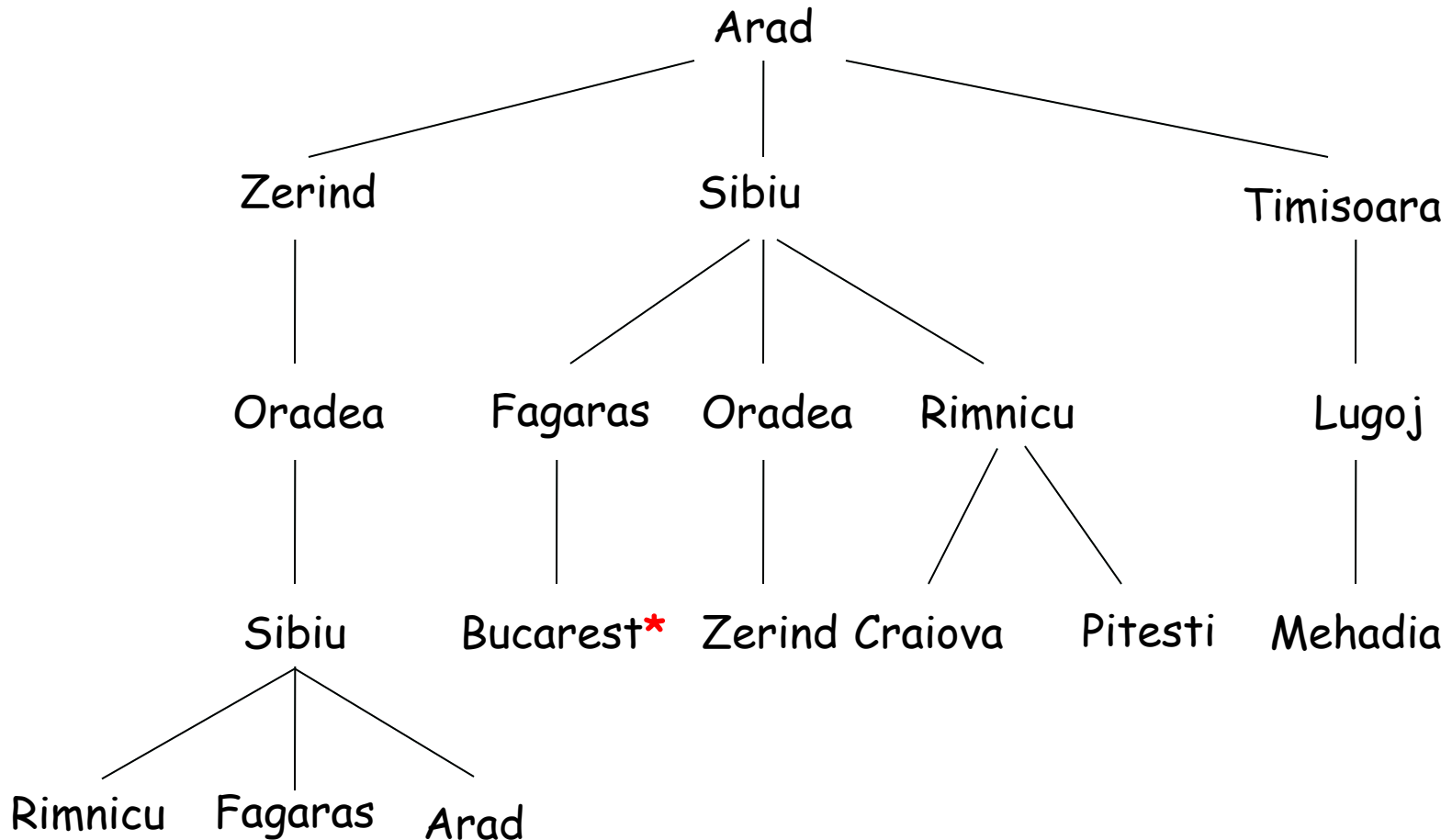
Búsqueda no informada

Aplicar búsqueda
preferente por amplitud
para ir de Arad a
Bucarest

Evite devolverse

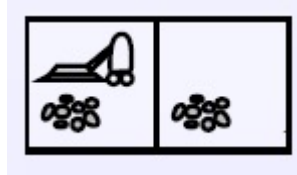


Búsqueda no informada



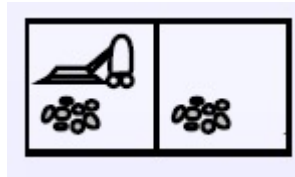
Búsqueda no informada

Aplicar búsqueda preferente
por amplitud

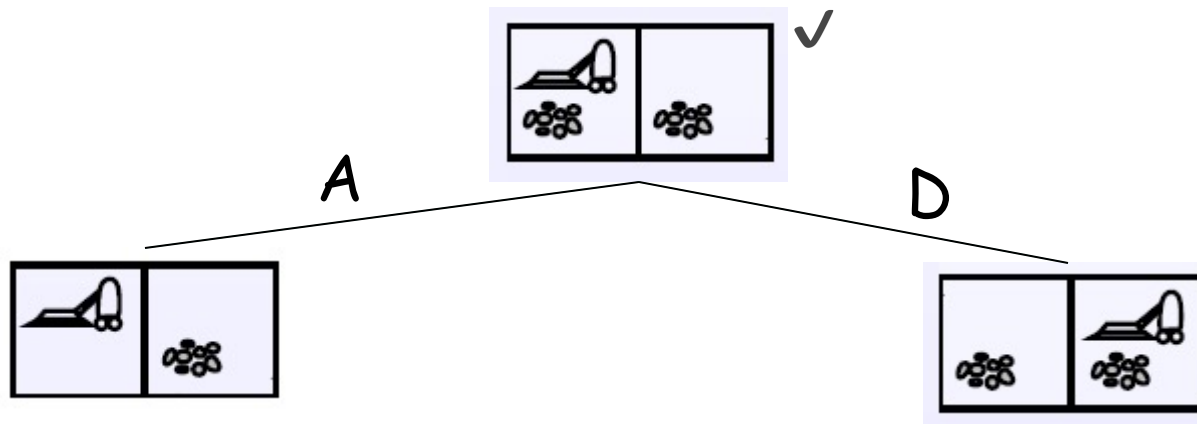


* No evite devolverse

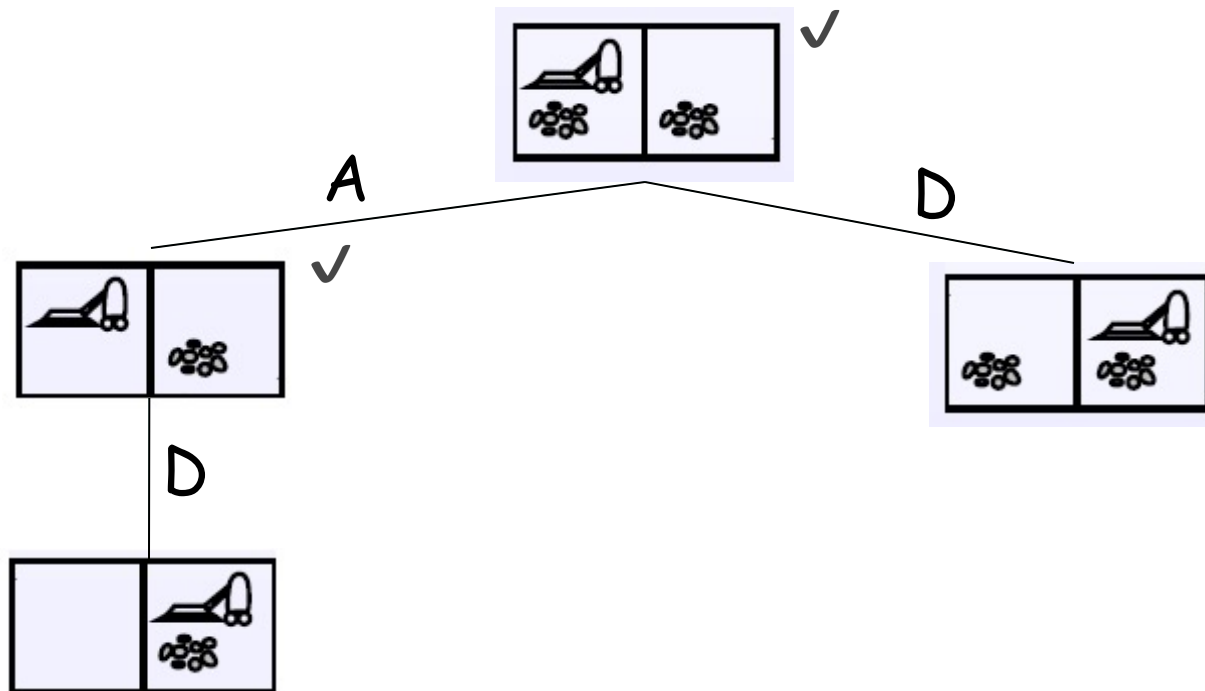
Búsqueda no informada



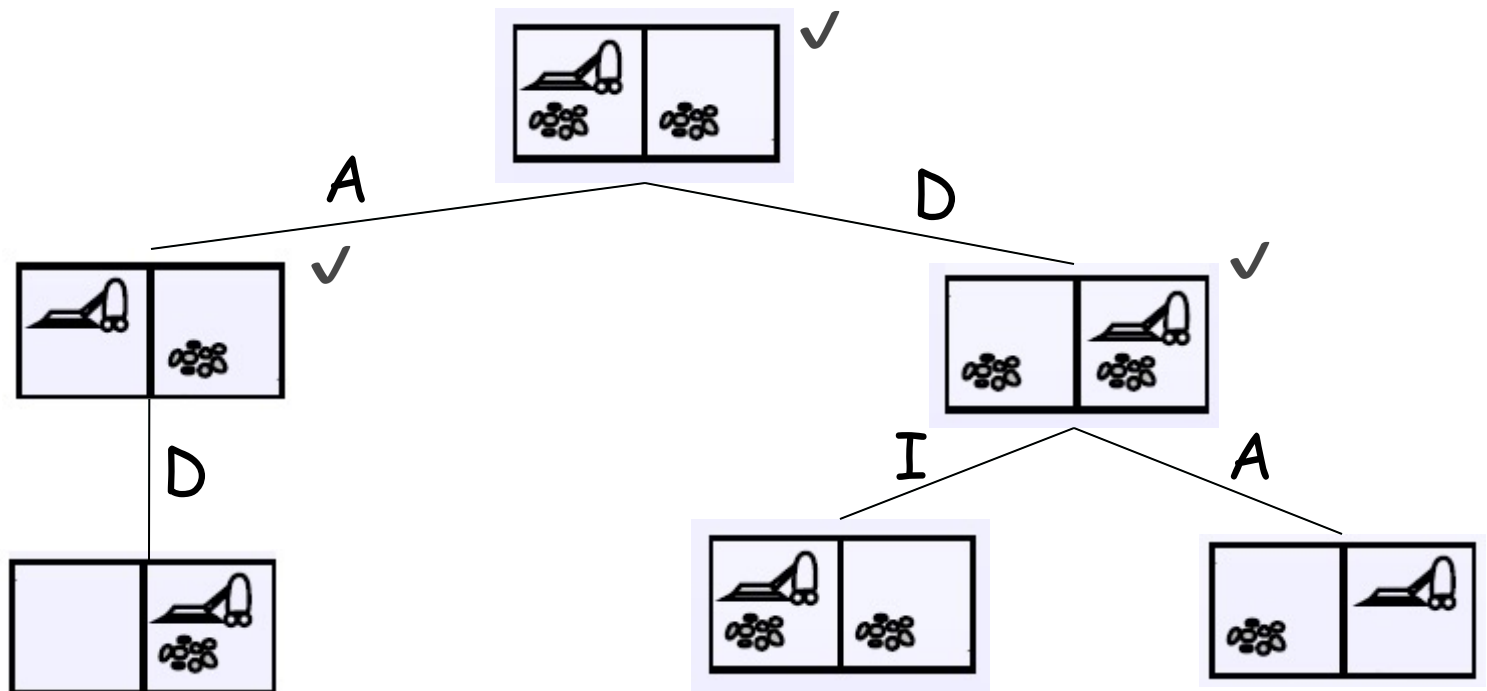
Búsqueda no informada



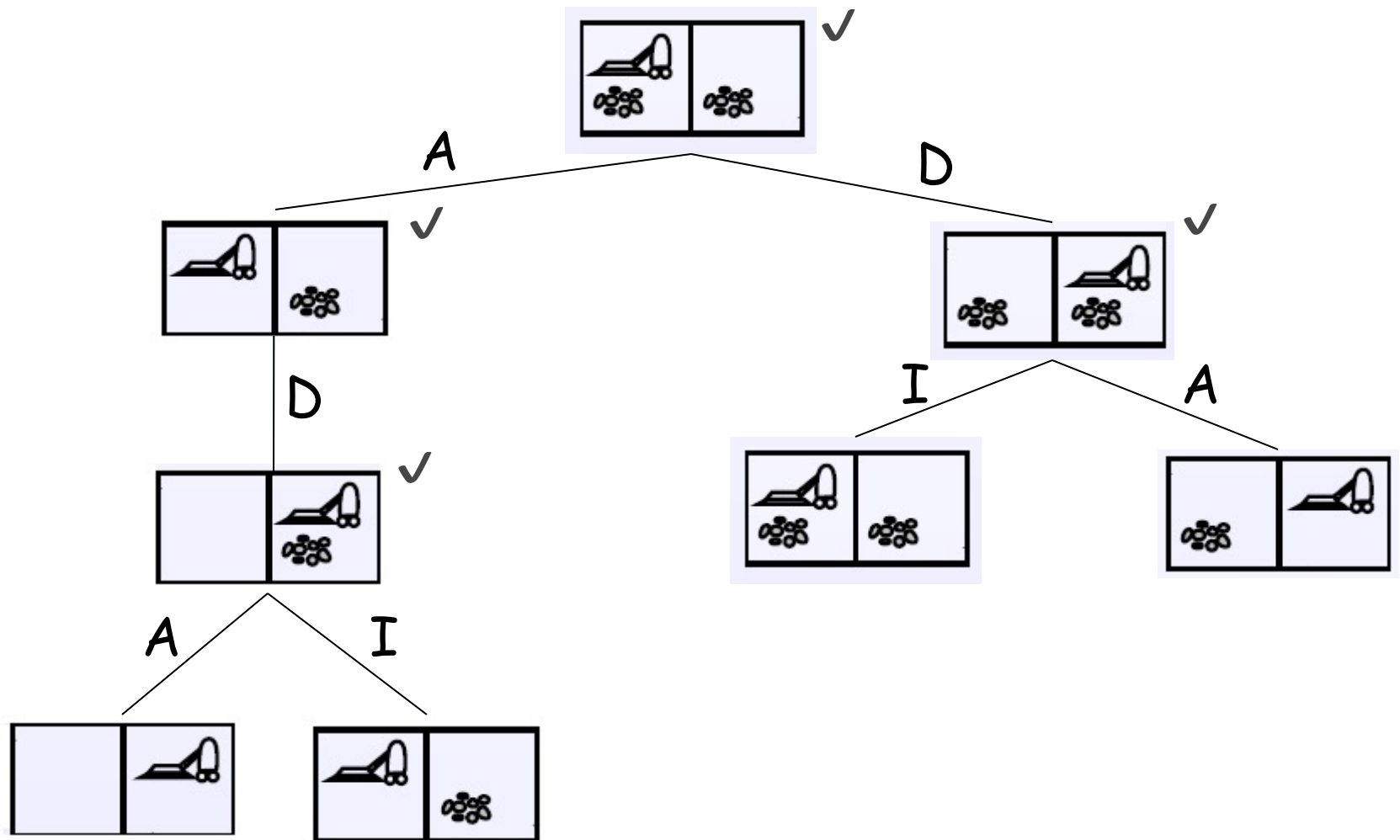
Búsqueda no informada



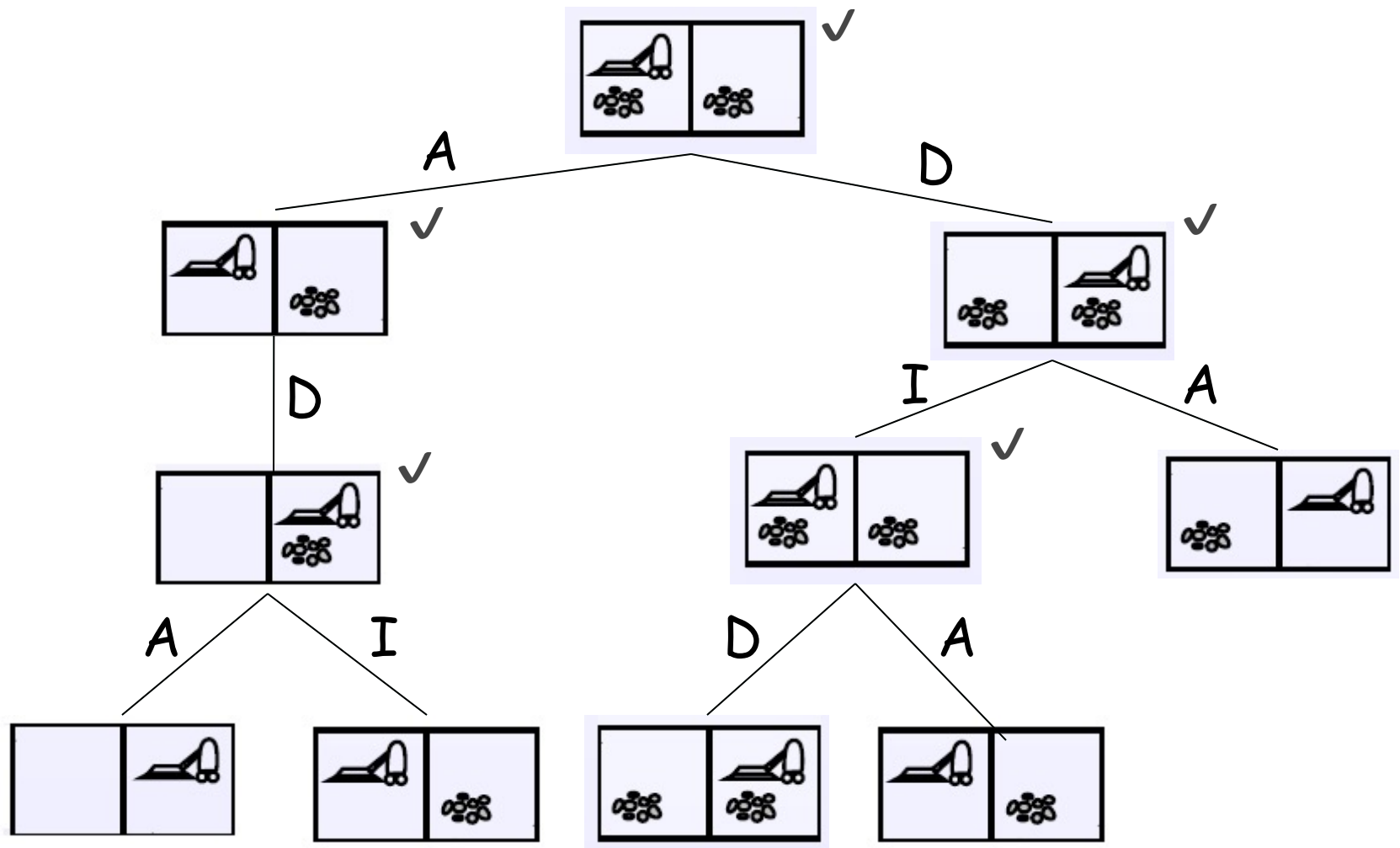
Búsqueda no informada



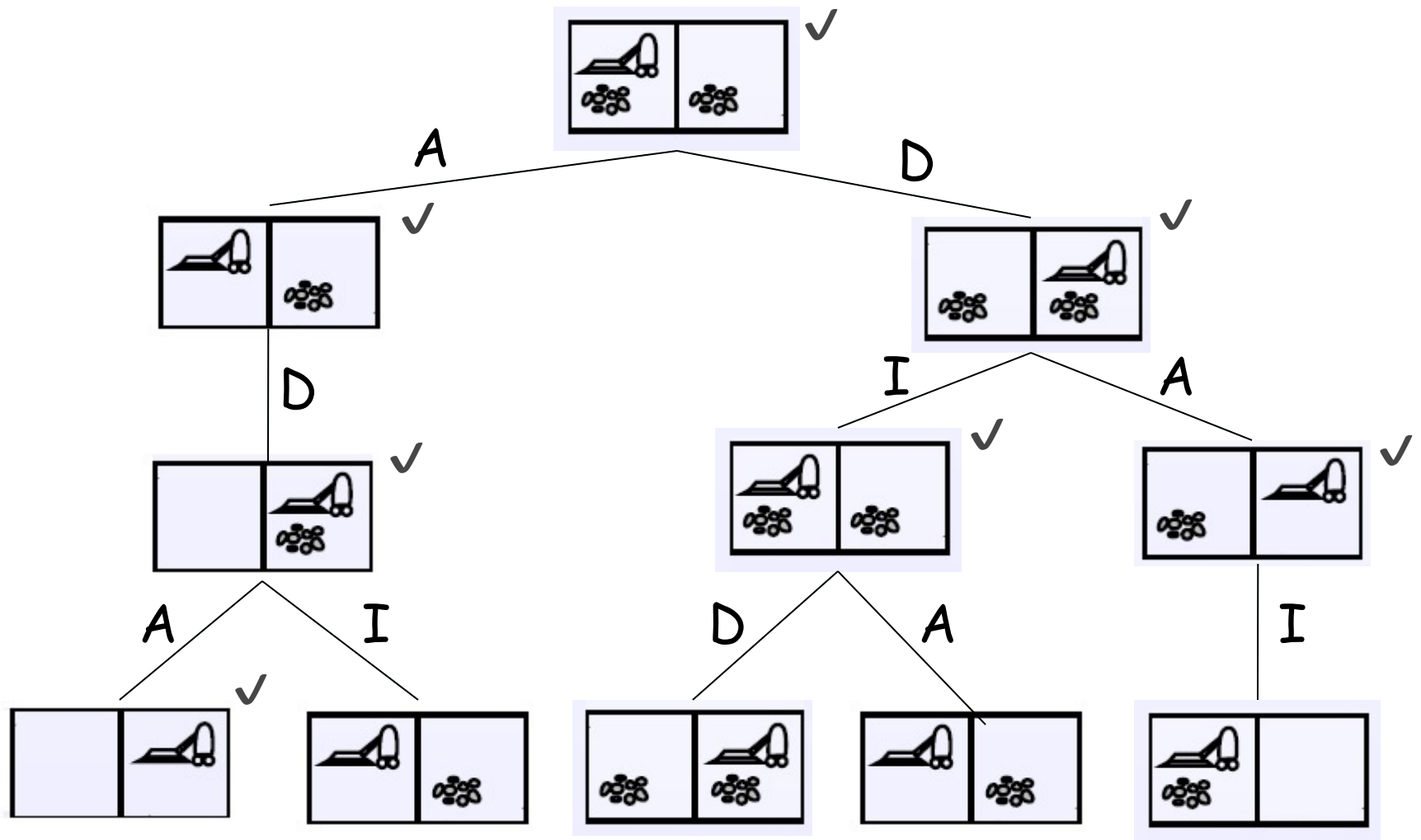
Búsqueda no informada



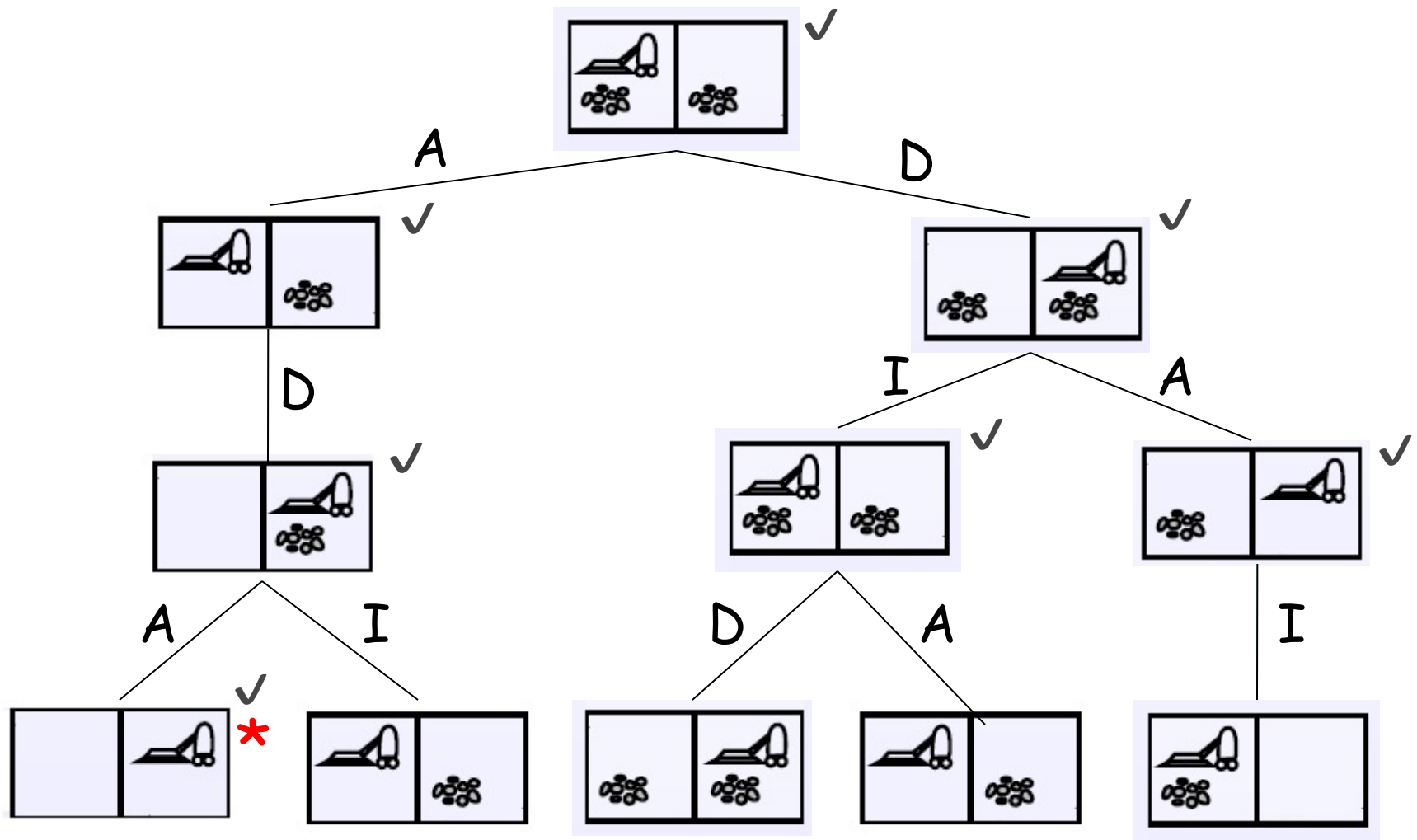
Búsqueda no informada



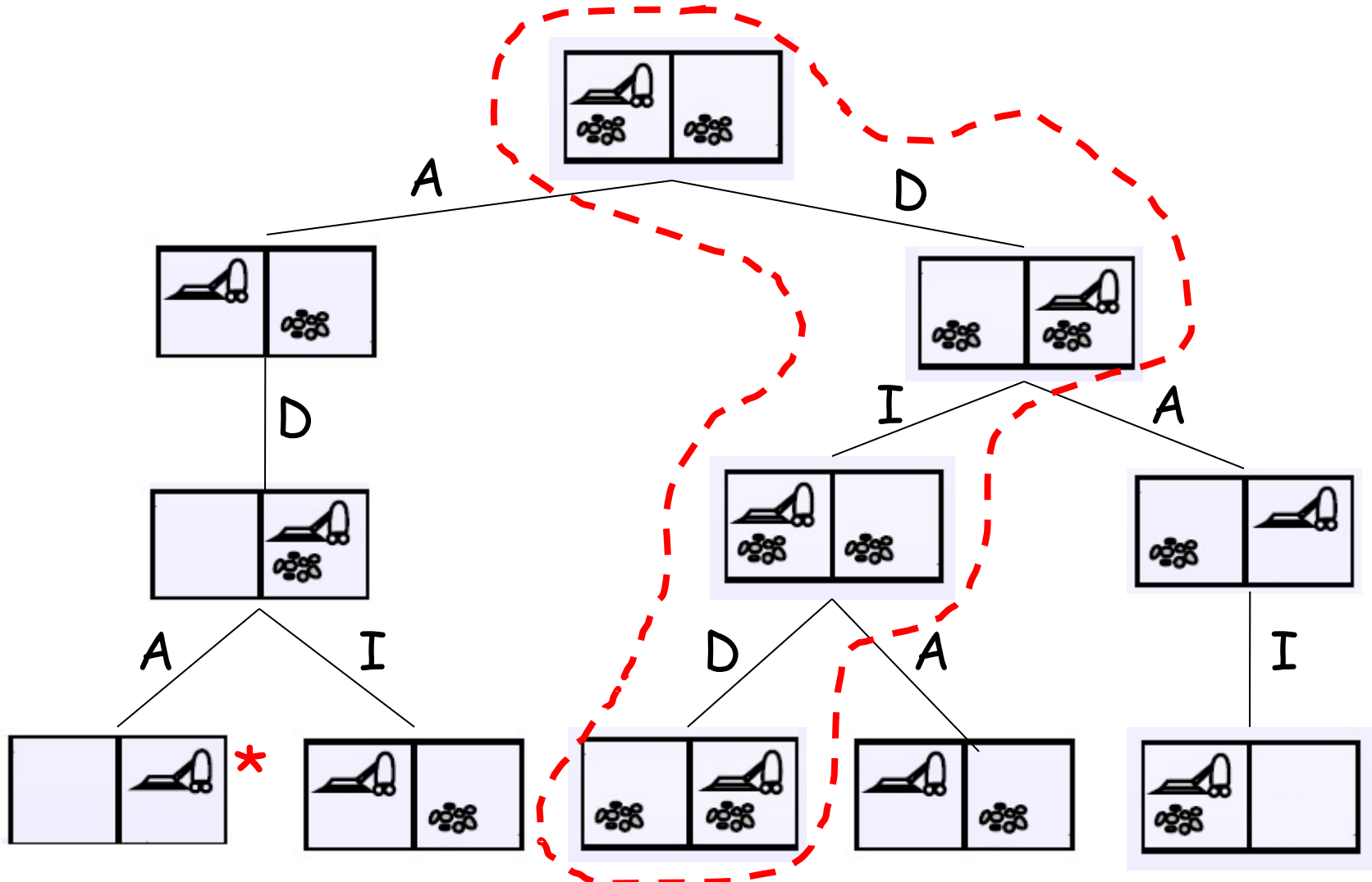
Búsqueda no informada



Búsqueda no informada



Búsqueda no informada



Los ciclos no afectan la búsqueda en amplitud

Búsqueda no informada

Aplicar búsqueda
preferente por amplitud

Evite devolverse

1		2
4	5	3
7	8	6

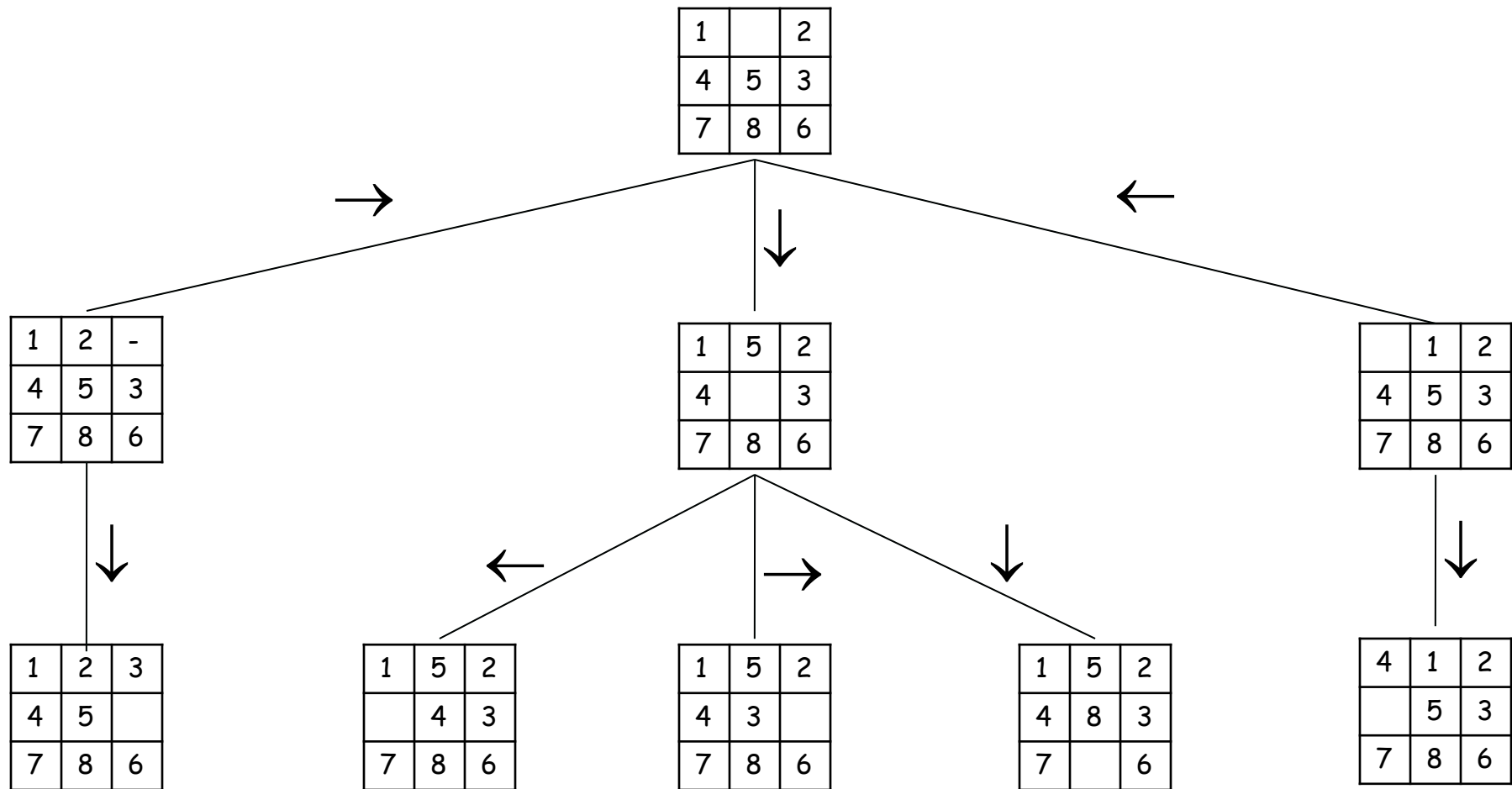
1		2
4	5	3
7	8	6

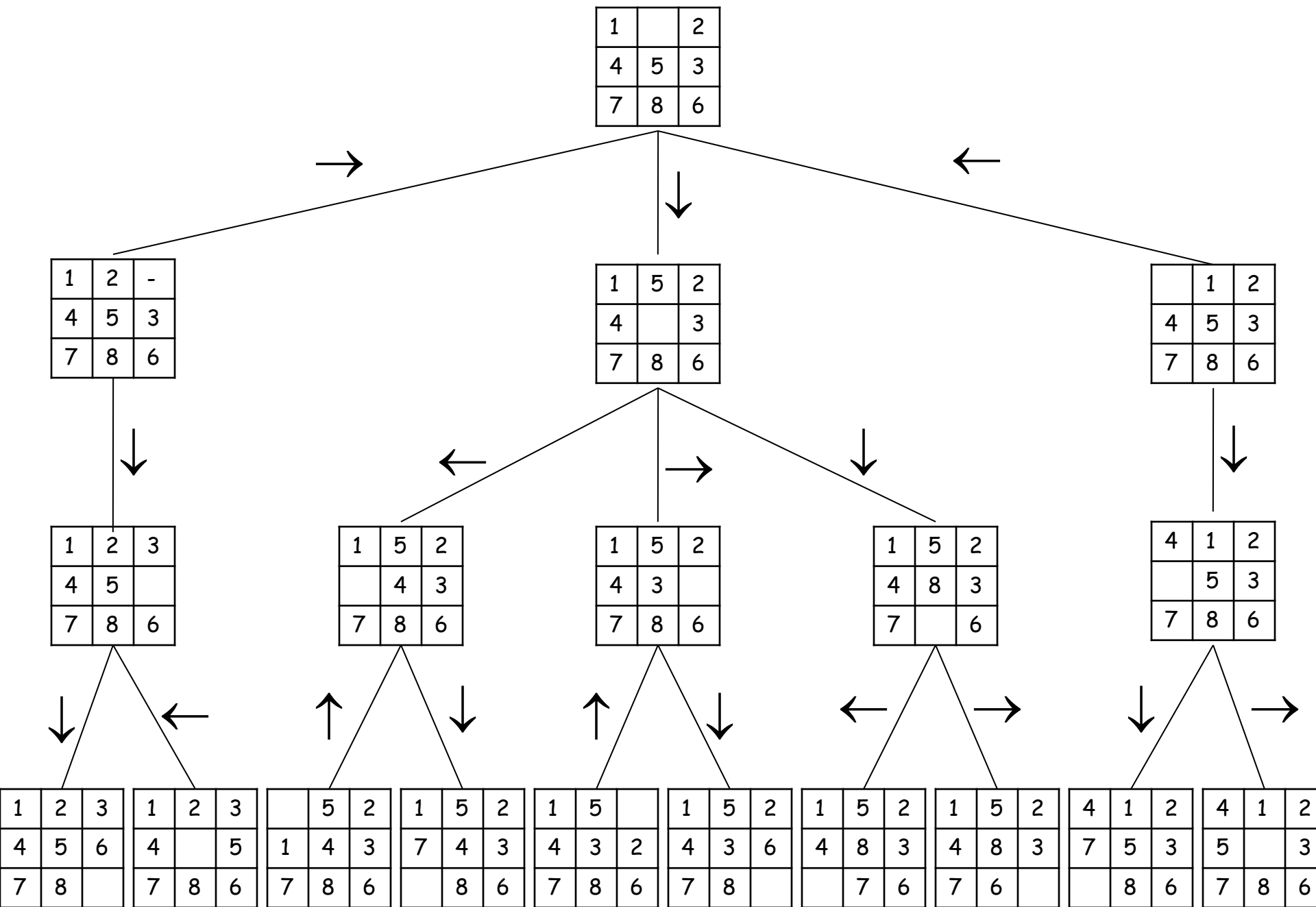


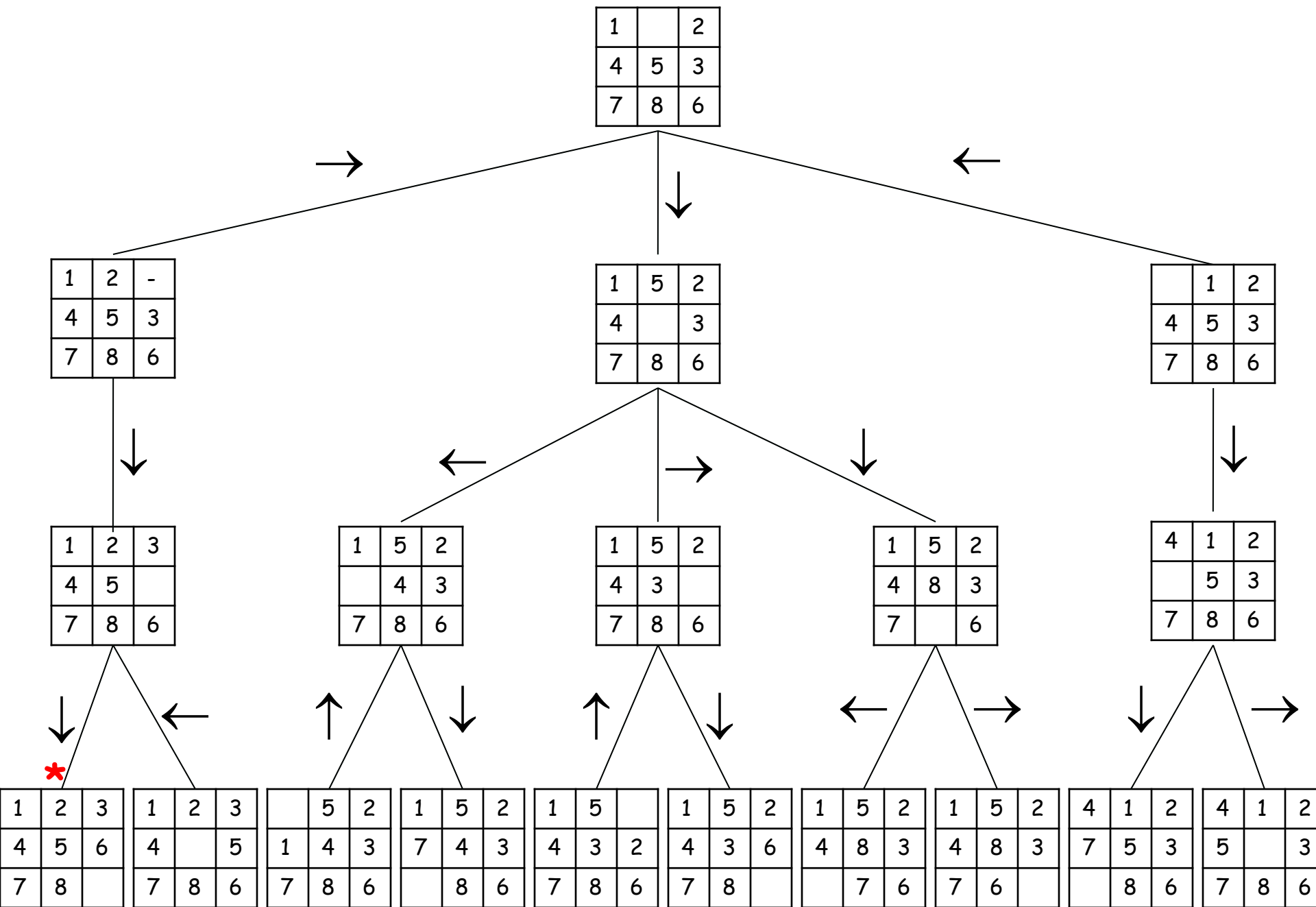
1	2	-
4	5	3
7	8	6

1	5	2
4		3
7	8	6

	1	2
4	5	3
7	8	6







function BUSQUEDA-PREFERENTE-POR-AMPLITUD(problema) returns solución o falla

Cola cola

cola = nodo_raiz(problema)

loop

if estaVacia(cola) then

return falló

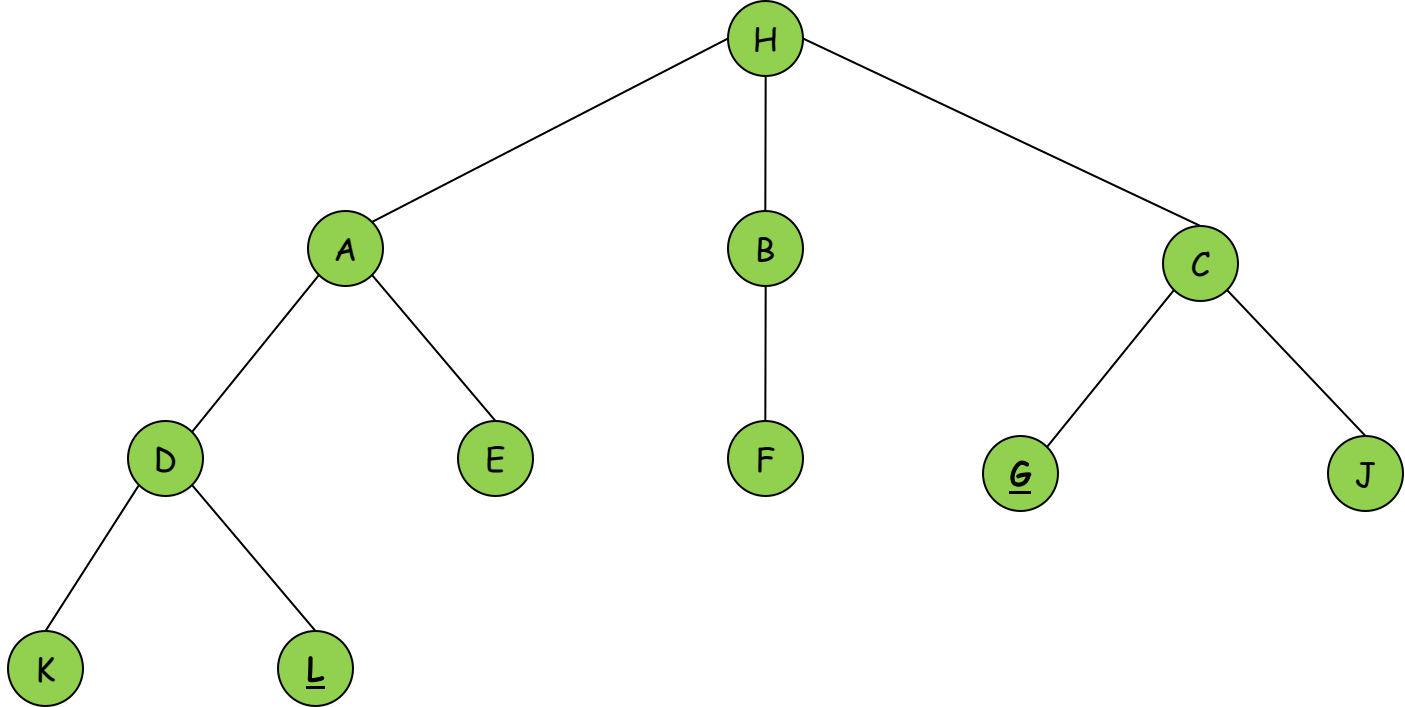
n=sacar(cola)

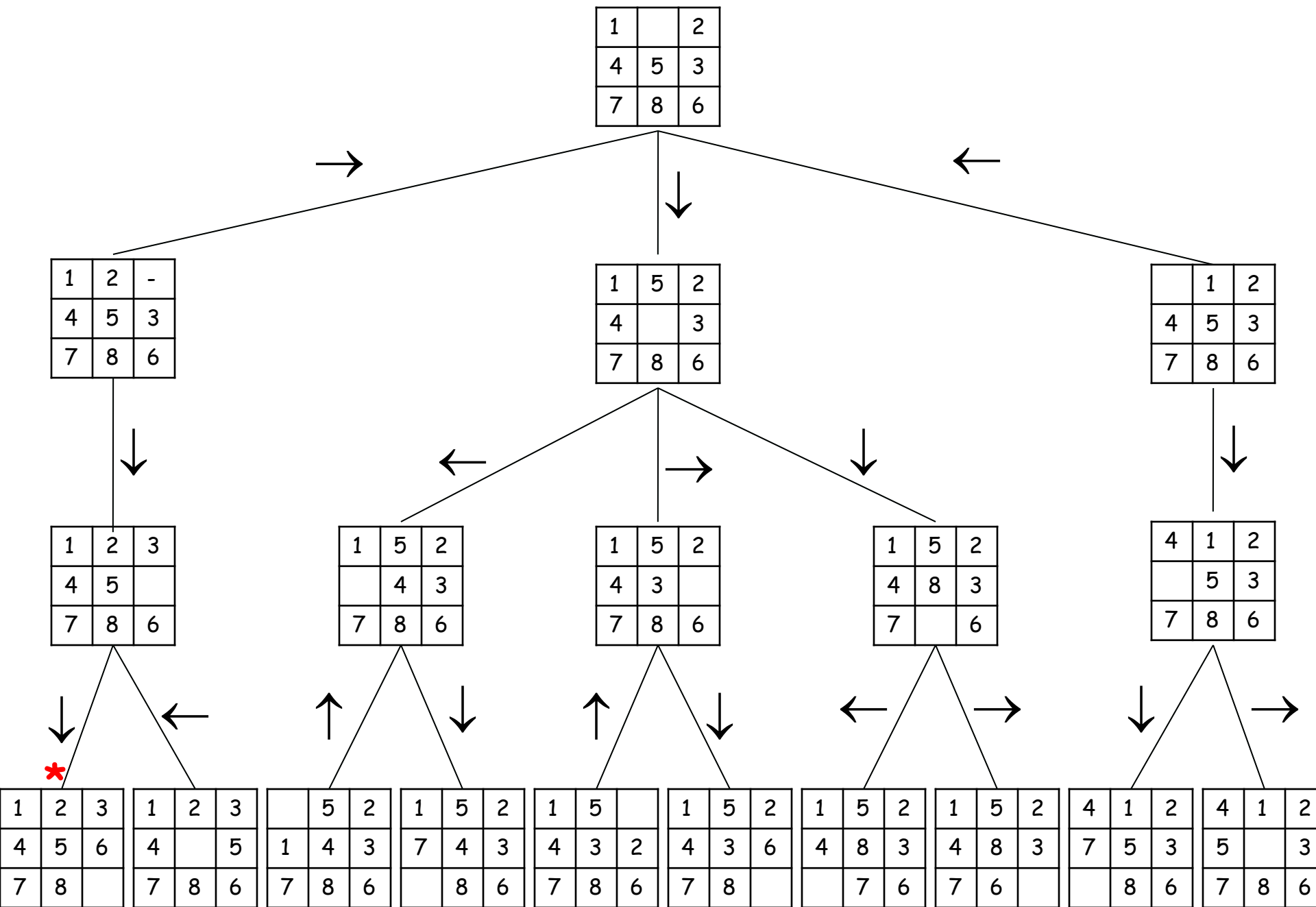
if n es un nodo meta then

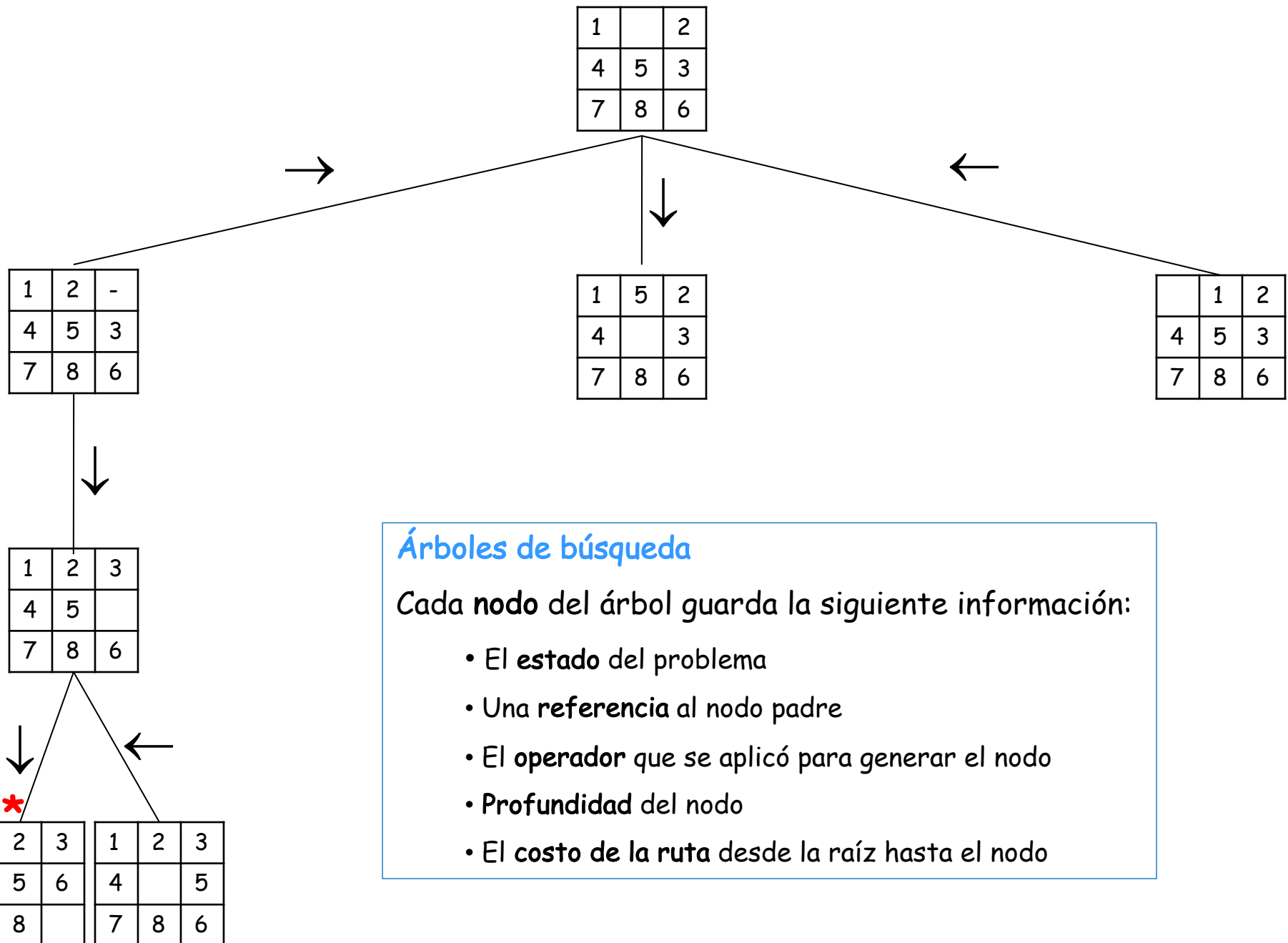
return encontró solución

expandir n y meter todos sus hijos al final de la cola

Estado de la cola											Iteración
H											
	A	B	C								Extraer H, meter hijos de H
		B	C	D	E						Extraer A, meter hijos de A
			C	D	E	F					Extraer B, meter hijos de B
				D	E	F	G	J			Extraer C, meter hijos de C
					E	F	G	J	K	L	Extraer D, meter hijos de D
						F	G	J	K	L	Extraer E
							G	J	K	L	Extraer F







1		2
4	5	3
7	8	6



1	2	-
4	5	3
7	8	6

1	5	2
4		3
7	8	6

	1	2
4	5	3
7	8	6



1	2	3
4	5	
7	8	6



1	2	3
4	5	6
7	8	

1	2	3
4		5
7	8	6

operador: ↓

1		2
4	5	3
7	8	6



1	2	-
4	5	3
7	8	6

1	5	2
4		3
7	8	6

	1	2
4	5	3
7	8	6



1	2	3
4	5	
7	8	6

operator: ↓



1	2	3
4	5	6
7	8	

1	2	3
4		5
7	8	6

operator: ↓

1		2
4	5	3
7	8	6



1	2	-
4	5	3
7	8	6

operador: →

1	5	2
4		3
7	8	6

	1	2
4	5	3
7	8	6



1	2	3
4	5	
7	8	6

operador: ↓



1	2	3
4	5	6
7	8	

1	2	3
4		5
7	8	6

operador: ↓

Búsqueda no informada

Búsqueda preferente por amplitud

- Completitud
- Complejidad temporal
- Complejidad espacial
- Solución óptima

Búsqueda no informada

Búsqueda preferente por amplitud

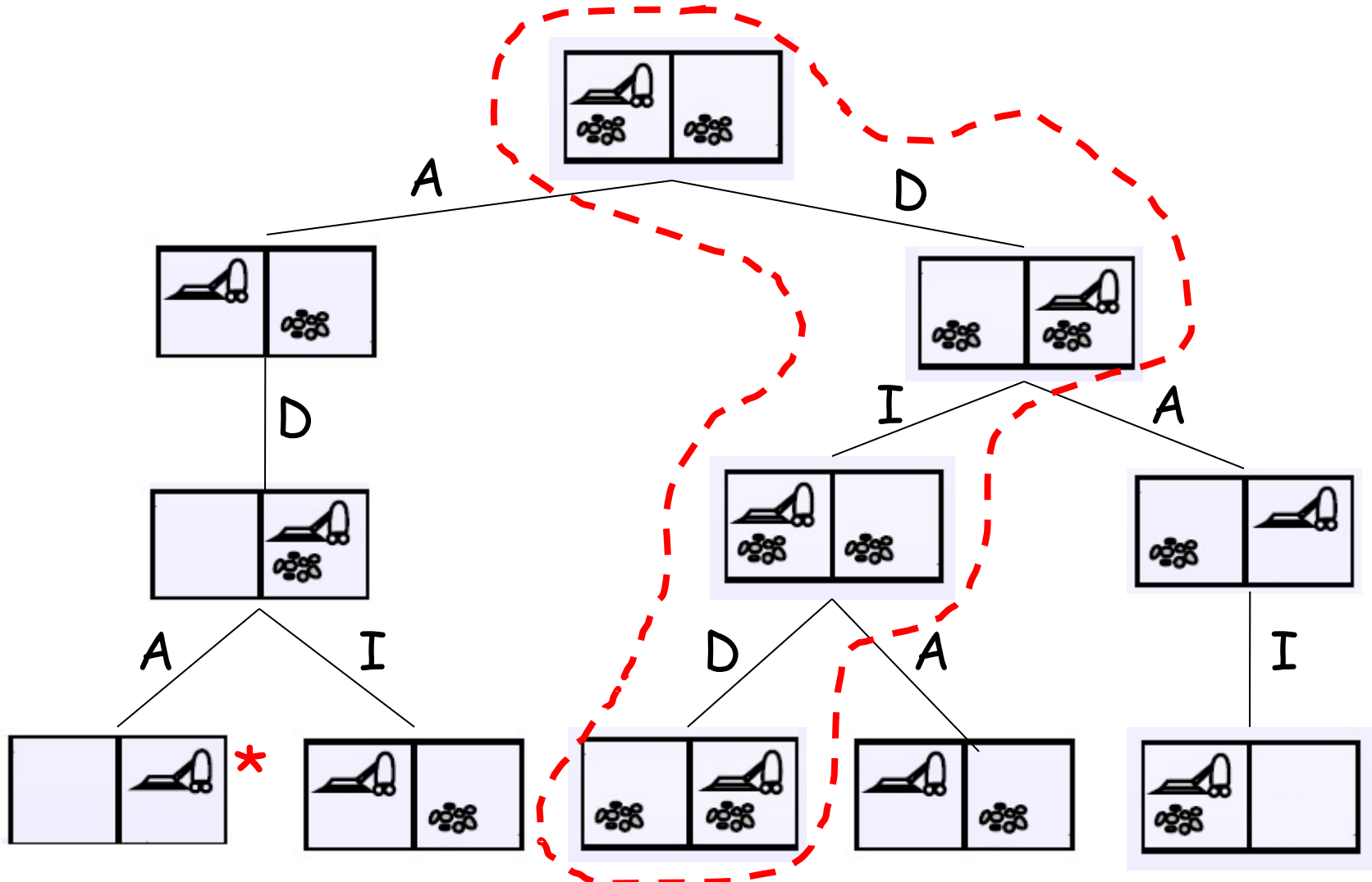
- **Completitud:** si existe una solución, ¿la estrategia la encuentra?
- **Complejidad temporal:** ¿cuánto tiempo se necesita para encontrar una solución?
- **Complejidad espacial:** ¿cuánta memoria se necesita para efectuar la búsqueda?
- **Solución óptima:** en caso de que existan varias soluciones, ¿se encuentra la mejor?

Búsqueda no informada

Búsqueda preferente por amplitud

- **Complejidad:** si existe una solución, la estrategia garantiza que se encontrará
- **Complejidad temporal**
- **Complejidad espacial**
- **Solución óptima**

Búsqueda no informada



Los ciclos no afectan la búsqueda en amplitud

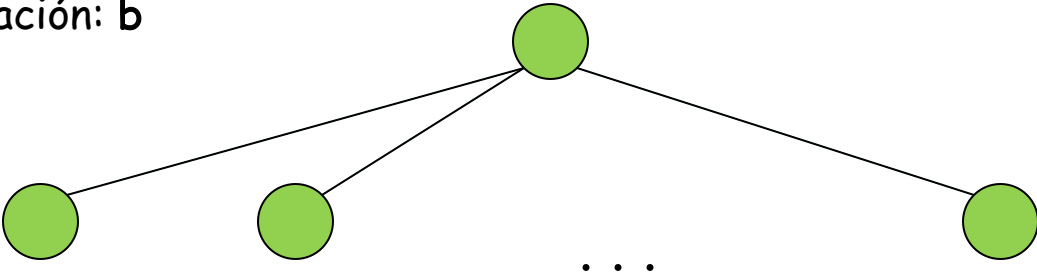
Búsqueda no informada

Búsqueda preferente por amplitud

- **Complejidad:** si existe una solución, ¿la estrategia la encuentra?
- **Complejidad temporal:** ¿cuánto tiempo se necesita para encontrar una solución?
- **Complejidad espacial:** ¿cuánta memoria se necesita para efectuar la búsqueda?
- **Solución óptima:** en caso de que existan varias soluciones, ¿se encuentra la mejor?



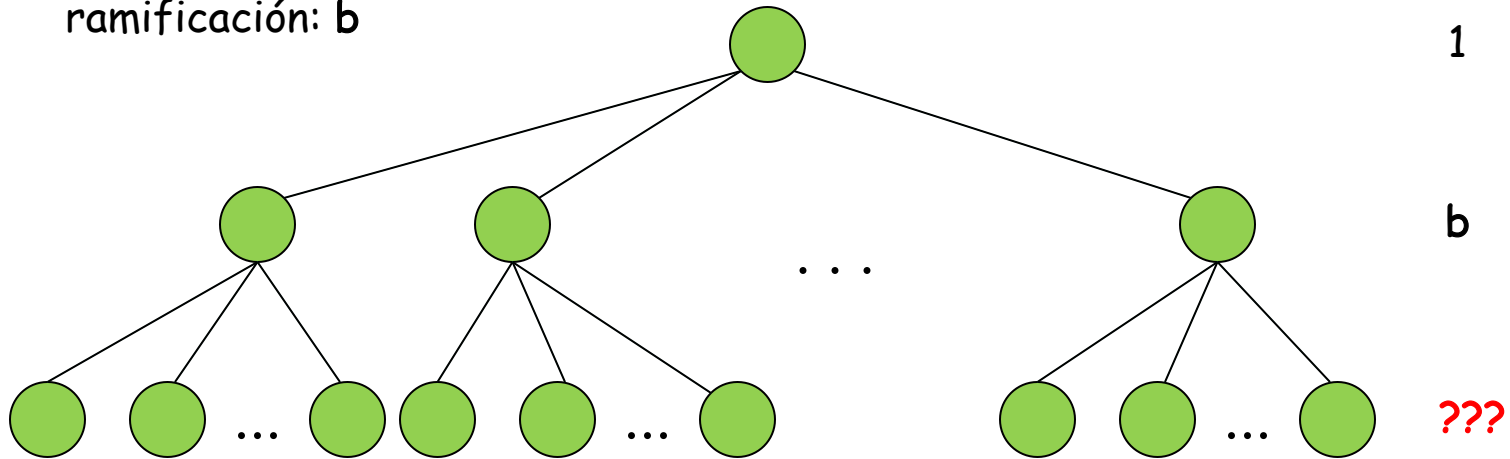
Factor de
ramificación: b



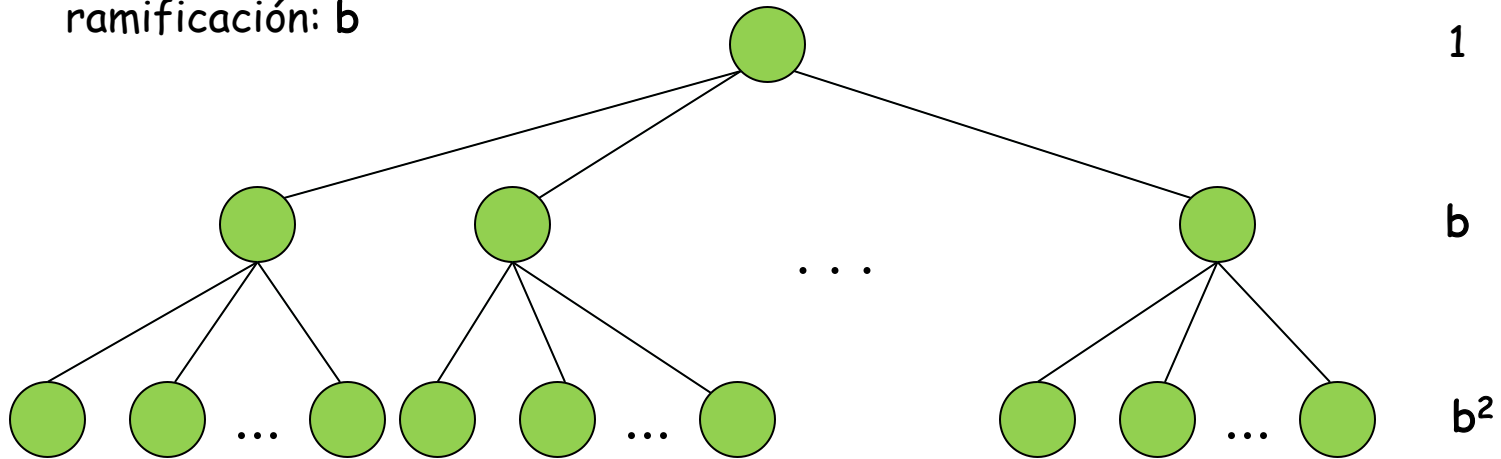
1

b

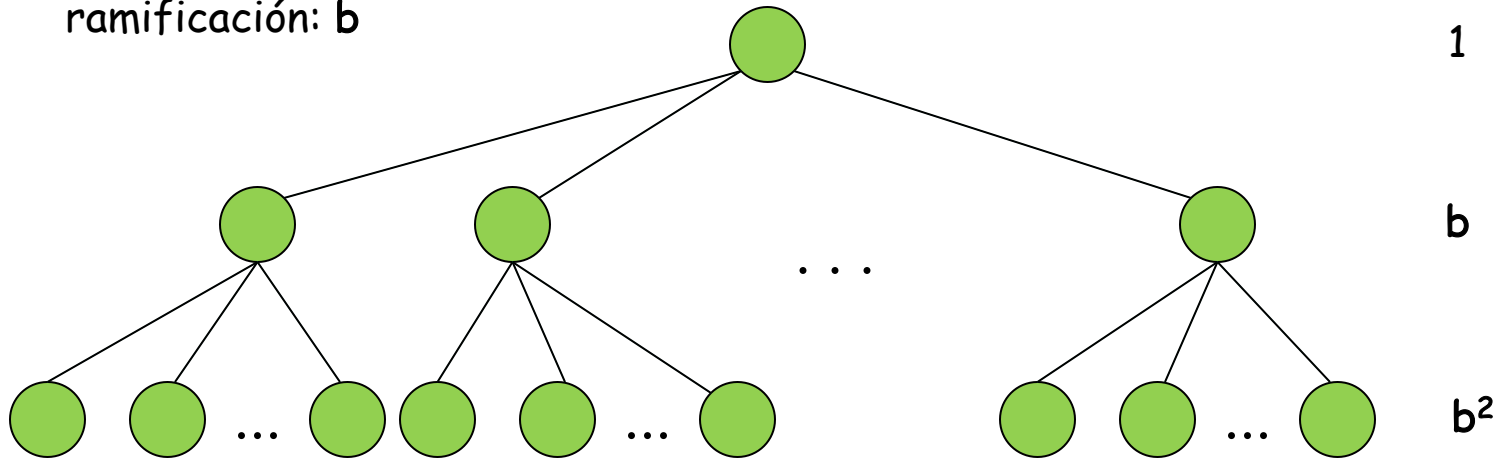
Factor de
ramificación: b



Factor de
ramificación: b

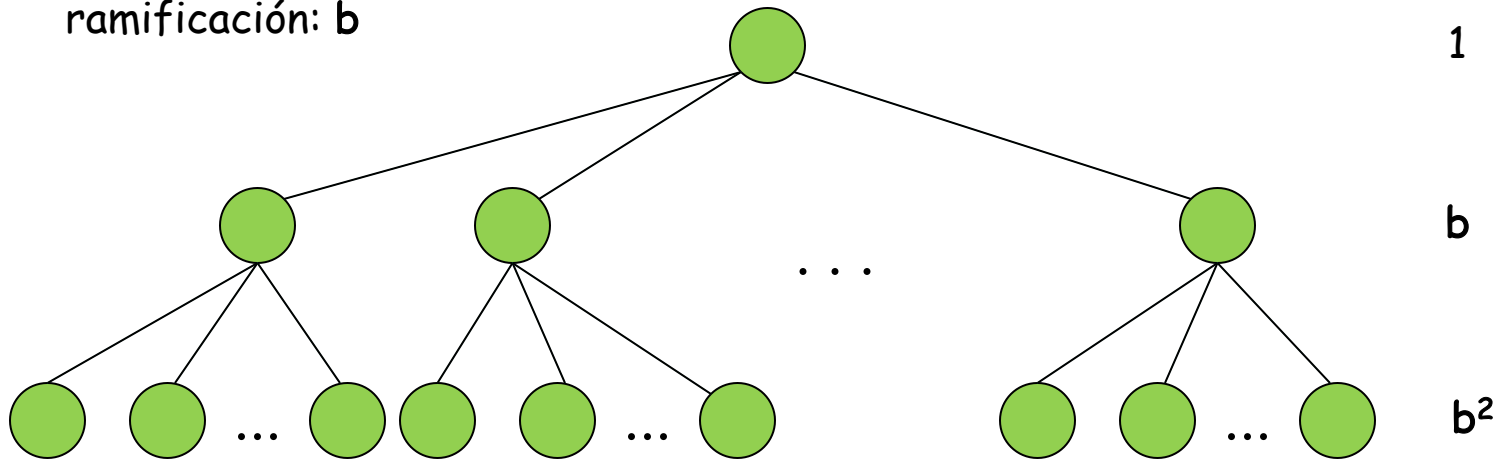


Factor de
ramificación: b



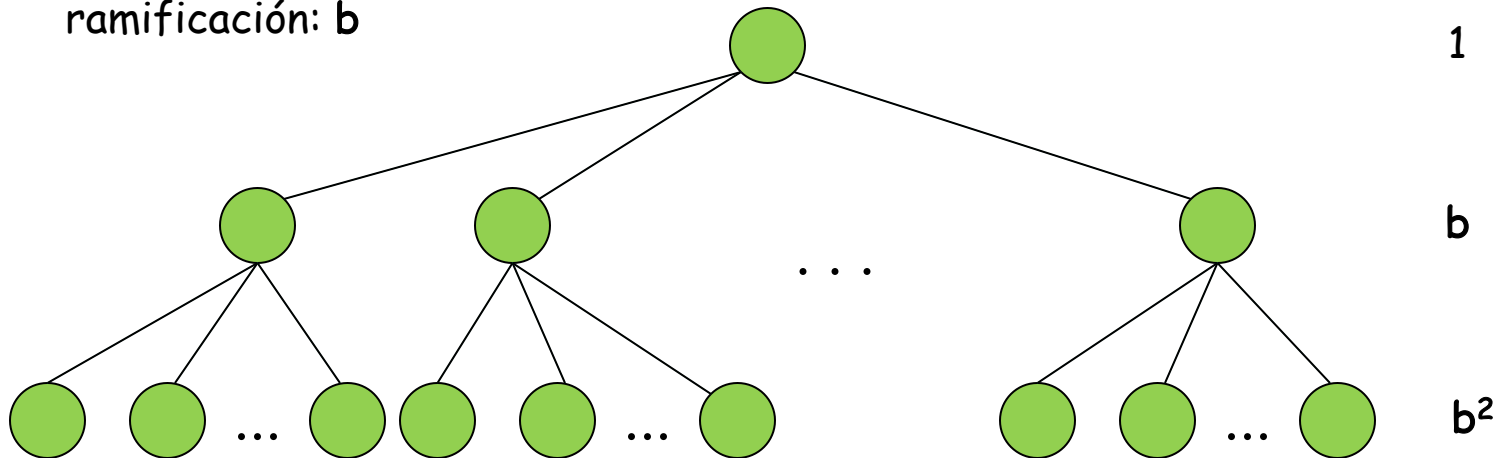
Cantidad máxima de nodos expandidos: $1 + b + b^2 + \dots + b^d$

Factor de
ramificación: b



Cantidad máxima de nodos expandidos: $1 + b + b^2 + \dots + b^d = O(b^d)$

Factor de
ramificación: b



Cantidad máxima de nodos expandidos: $1 + b + b^2 + \dots + b^d = O(b^d)$

$b=10$

Profundidad	Nodos	Tiempo
2	111	0.1 segundos
4	11111	11 segundos
6	10^6	18 minutos
8	10^8	31 horas
10	10^{10}	128 días
12	10^{12}	35 años
14	10^{14}	35000 años

Suponiendo
expansión de 1000
nodos por segundo

Búsqueda no informada

Búsqueda preferente por amplitud

- **Complejidad:** si existe una solución la estrategia garantiza que se encontrará
- **Complejidad temporal:** $O(b^d)$
- **Complejidad espacial**
- **Solución óptima**




Búsqueda no informada

Búsqueda preferente por amplitud

- **Completitud:** si existe una solución la estrategia garantiza que se encontrará
- **Complejidad temporal:** $O(b^d)$
- **Complejidad espacial:** cada nodo del árbol se debe guardar en memoria, la complejidad espacial también es $O(b^d)$
- **Solución óptima**




Búsqueda no informada

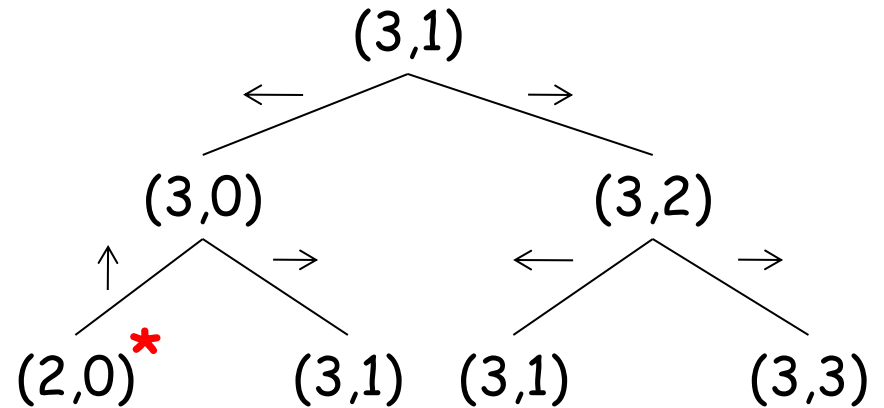
Aplicar búsqueda preferente
por amplitud

	0	1	2	3
0				
1				
2				
3				

Búsqueda no informada

Aplicar búsqueda preferente por amplitud

	0	1	2	3
0				
1				
2				
3				



Búsqueda no informada

Búsqueda preferente por amplitud

- **Completitud:** si existe una solución la estrategia garantiza que se encontrará
- **Complejidad temporal:** $O(b^d)$
- **Complejidad espacial:** cada nodo del árbol se debe guardar en memoria, la complejidad espacial también es $O(b^d)$
- **Solución óptima:** no

Búsqueda no informada

Búsqueda no informada:

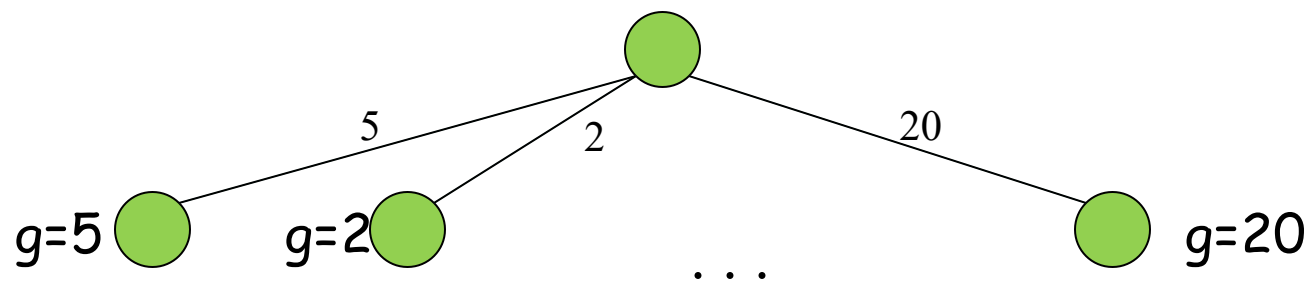
- Preferente por amplitud
- De costo uniforme
- Preferente por profundidad
- Limitada por profundidad
- Por profundización iterativa

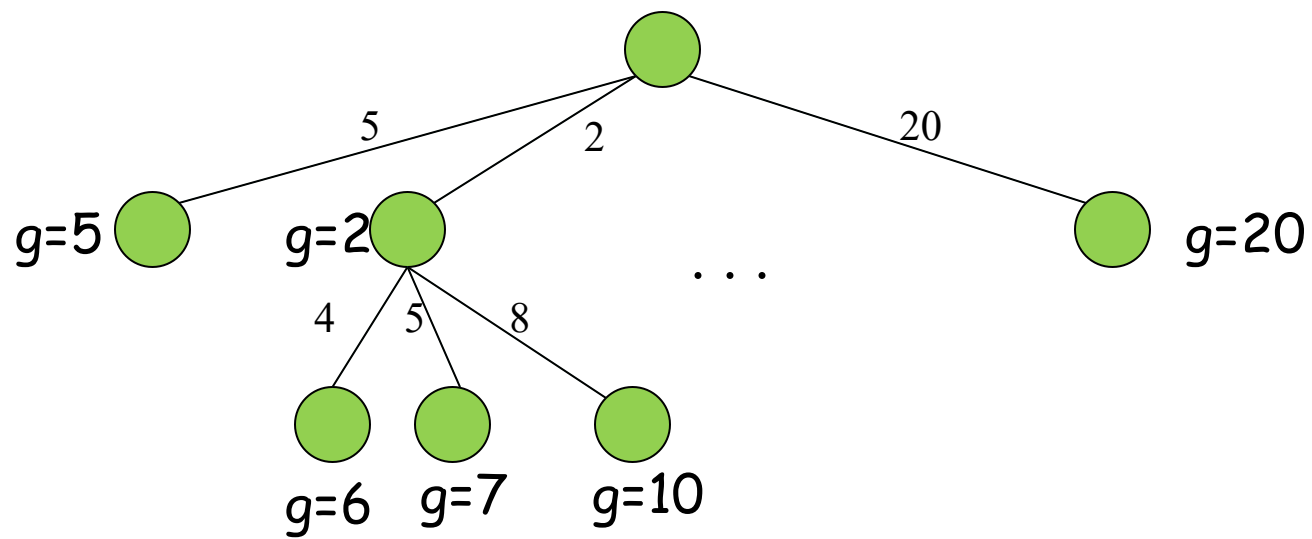
Búsqueda no informada

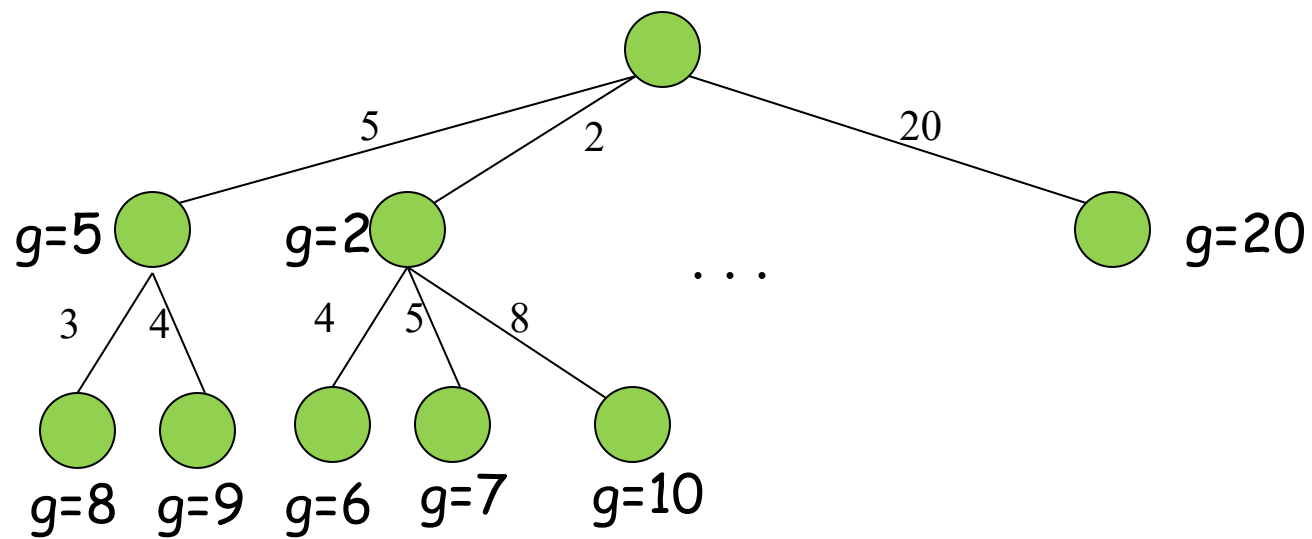
Búsqueda de costo uniforme

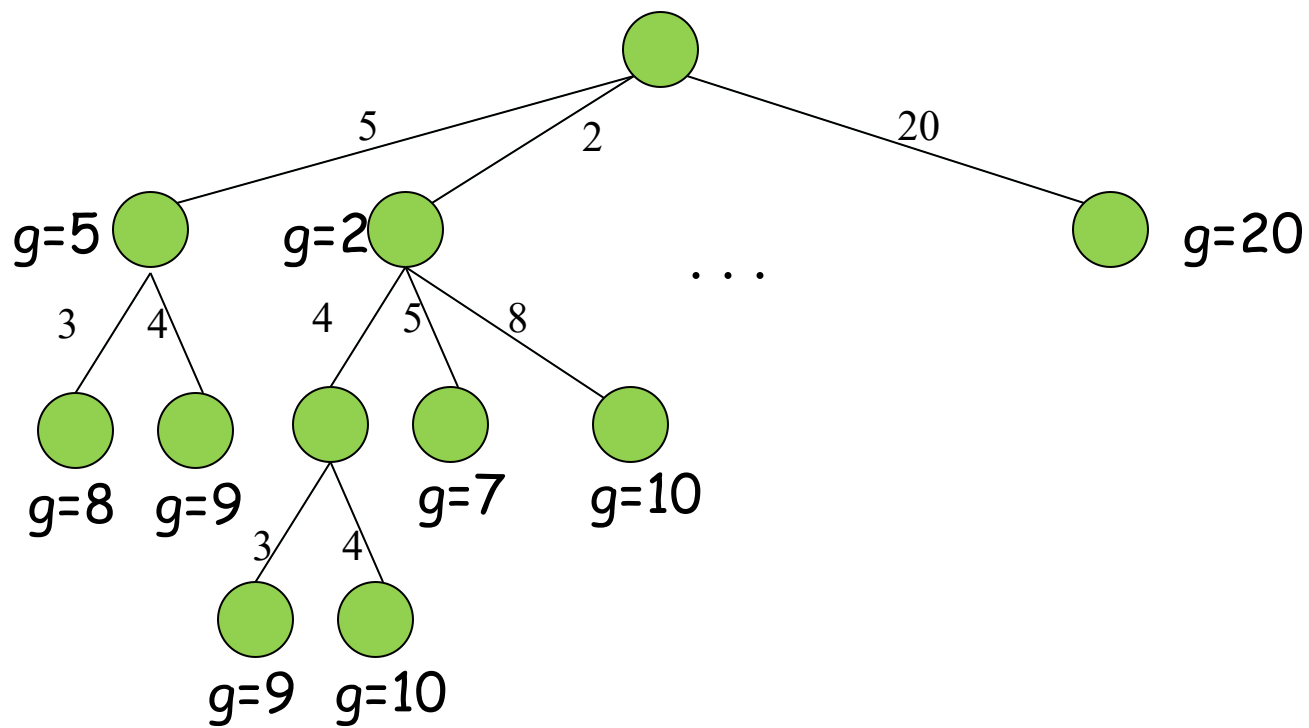
- Se expande el nodo raíz
- En cada nodo n del árbol se calcula el costo de ruta $g(n)$
- Se expande el nodo de menor costo, sin importar a qué profundidad esté

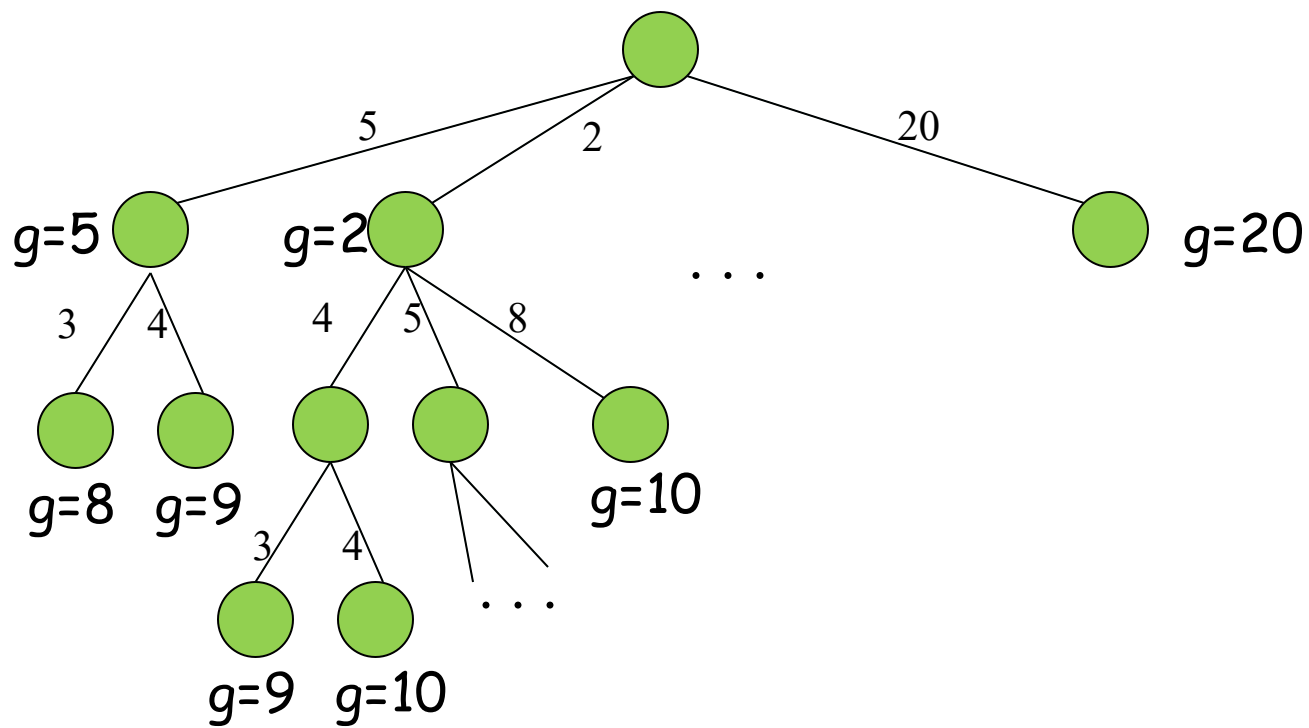








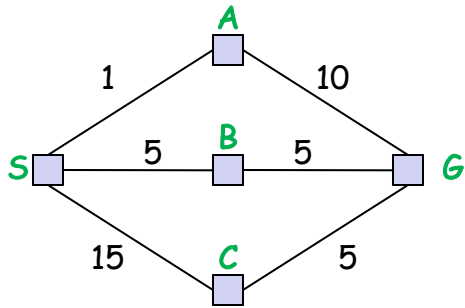




Búsqueda no informada

Búsqueda de costo uniforme

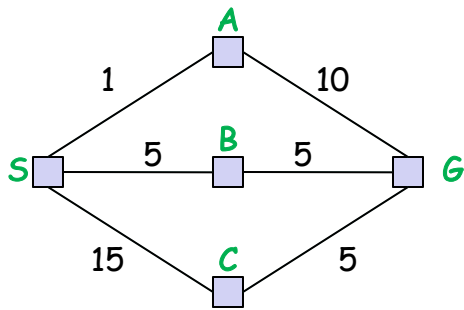
Problema: ir de S a G



Búsqueda no informada

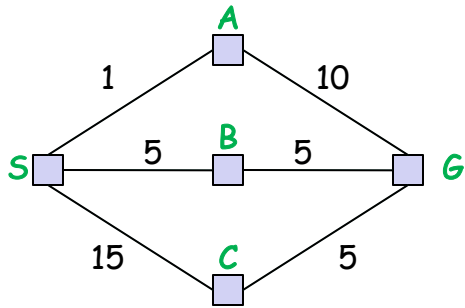
Búsqueda de costo uniforme

Se utiliza $g(n)$ como la suma de costos para llegar al nodo n a través de un camino dado



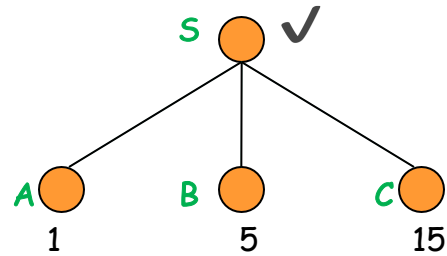
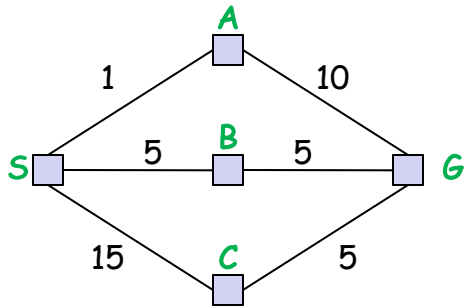
Búsqueda no informada

Búsqueda de costo uniforme



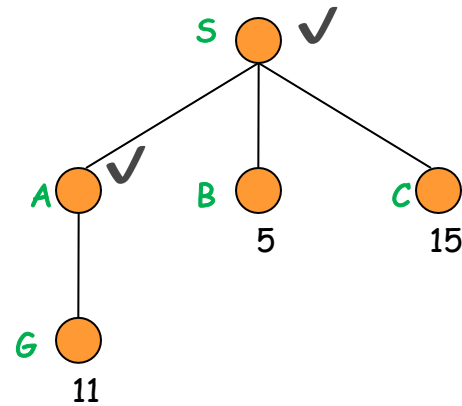
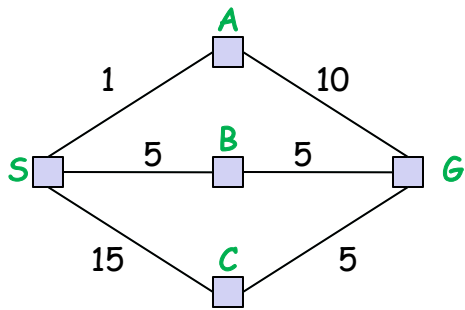
Búsqueda no informada

Búsqueda de costo uniforme



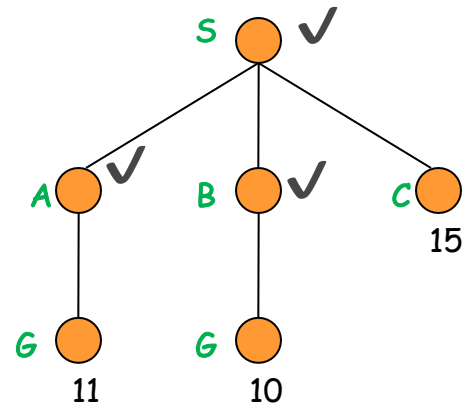
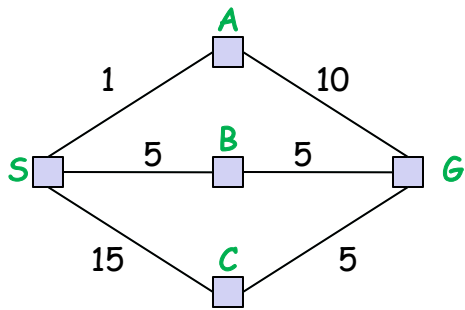
Búsqueda no informada

Búsqueda de costo uniforme



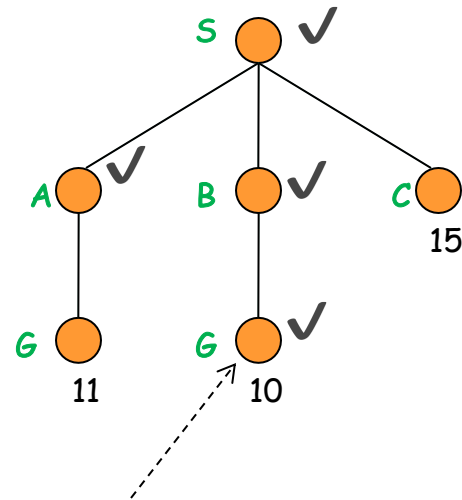
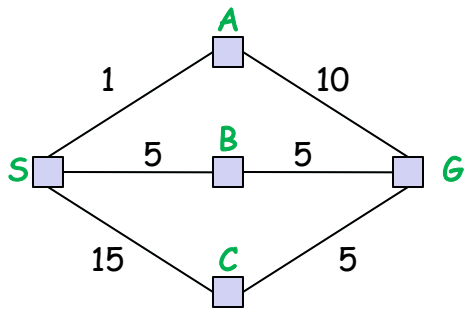
Búsqueda no informada

Búsqueda de costo uniforme



Búsqueda no informada

Búsqueda de costo uniforme

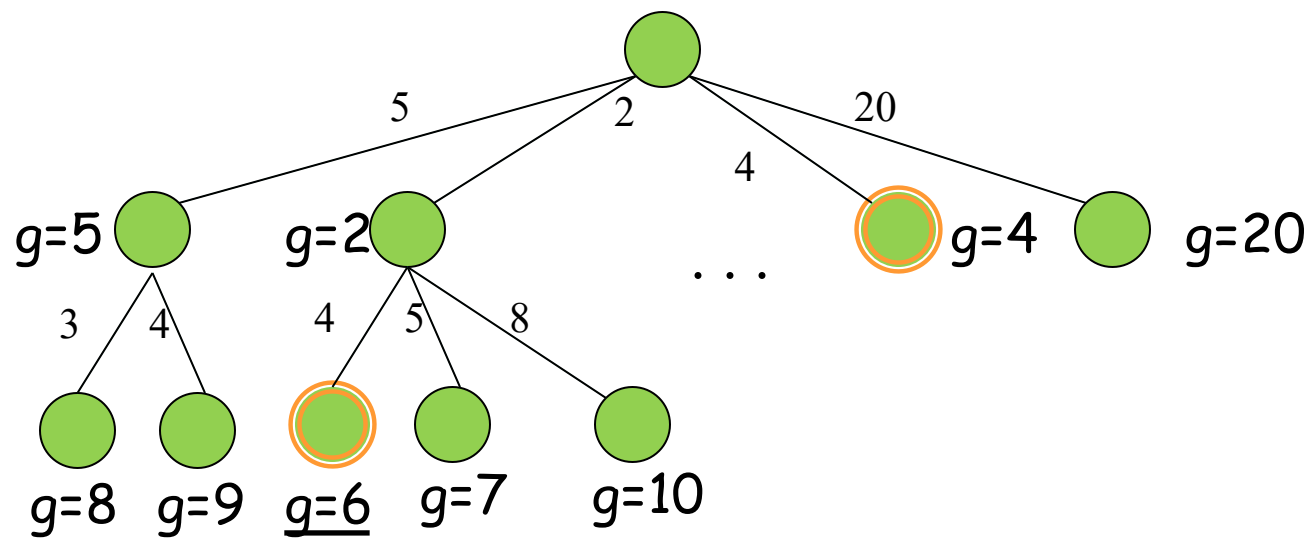


Se expande el nodo y se reconoce SBG como la solución

Búsqueda no informada

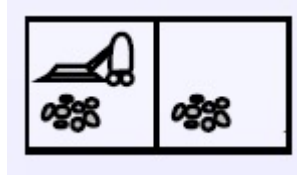
Búsqueda de costo uniforme

- La primera solución encontrada será la más barata puesto que si hubiera una más barata habría sido expandida anteriormente



Búsqueda no informada

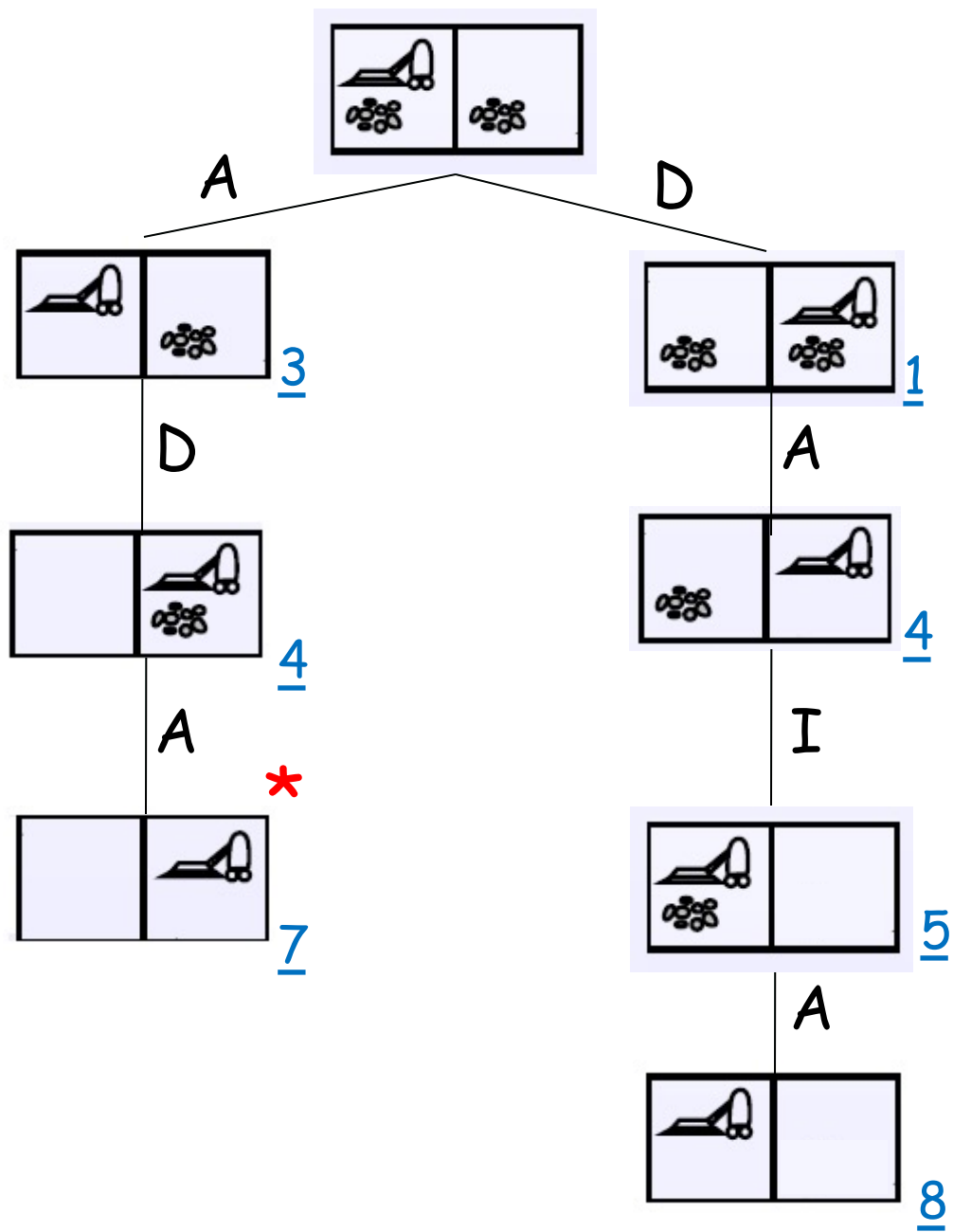
Aplicar búsqueda de costo uniforme



Costos

- Aspirar: 3
- Moverse a la izquierda: 1
- Moverse a la derecha: 1

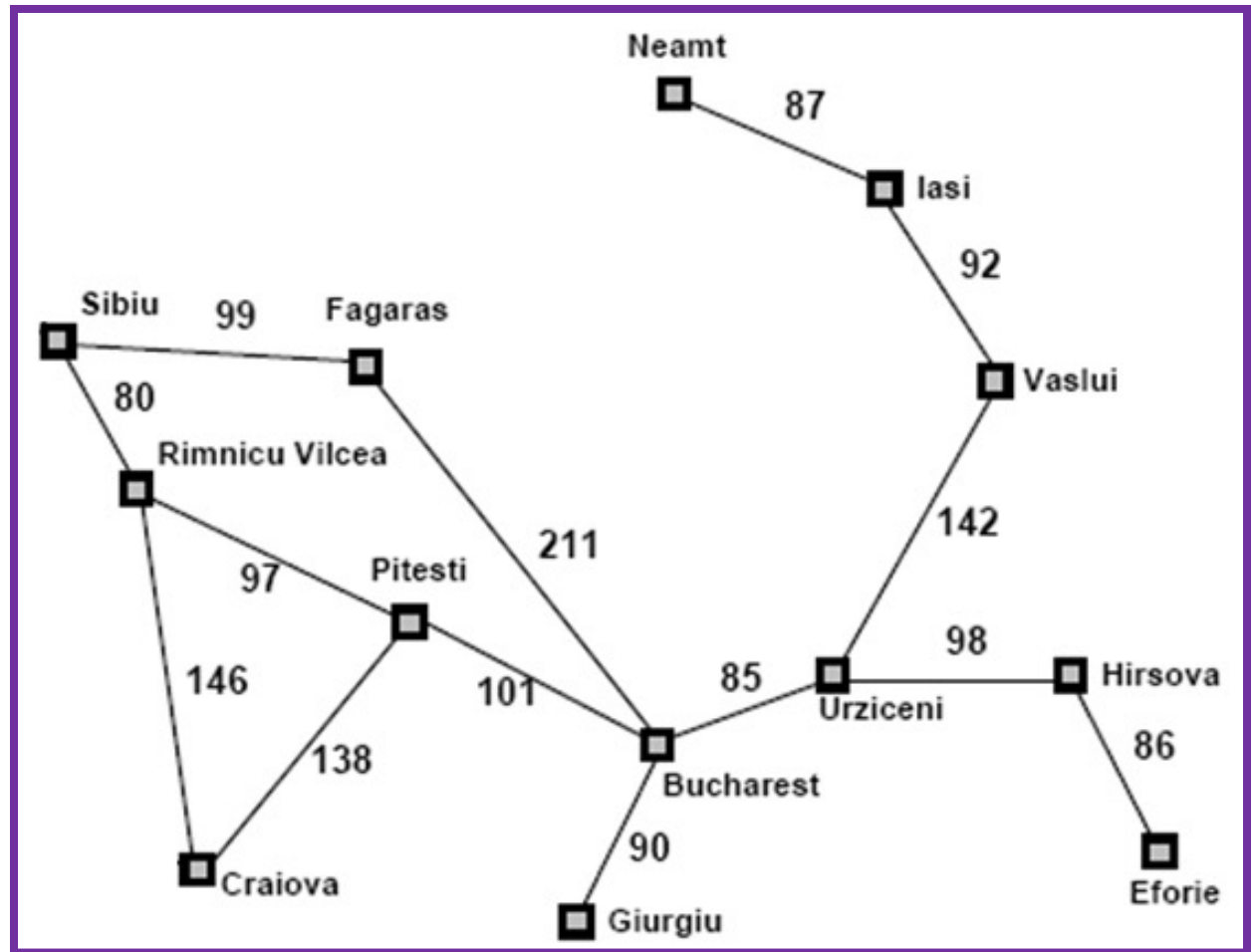
* Evite devolverse



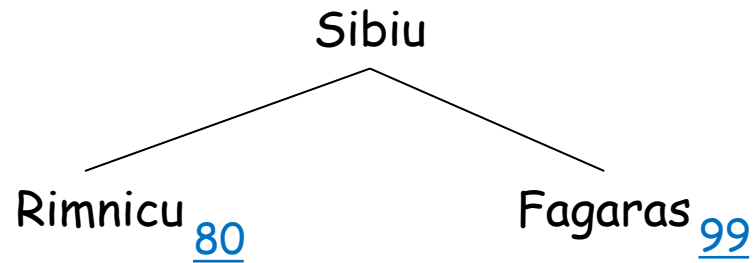
Búsqueda no informada

Aplicar búsqueda de costo uniforme
para ir de Sibiu a Bucarest

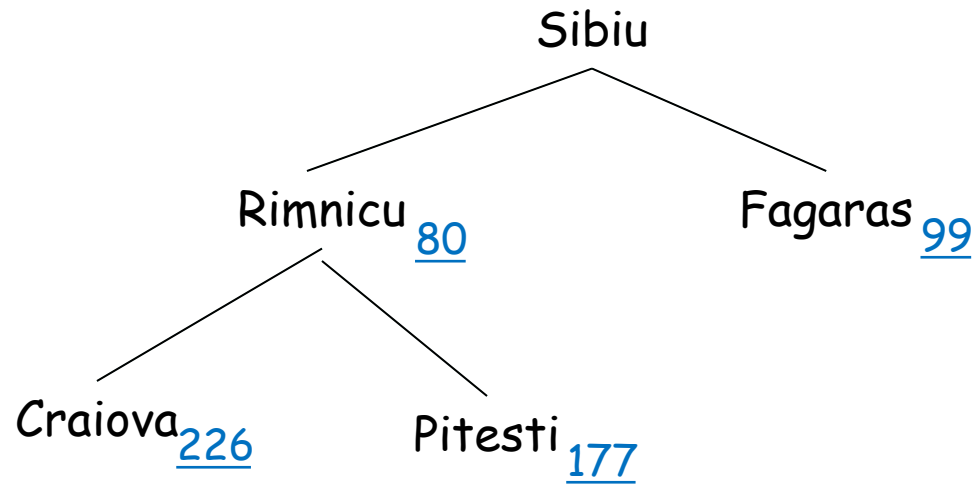
Evite devolverse



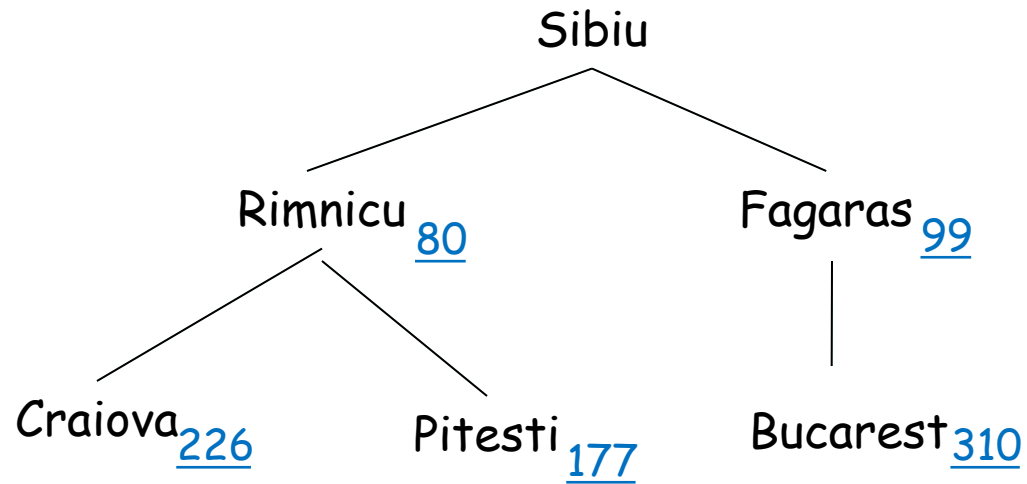
Búsqueda no informada



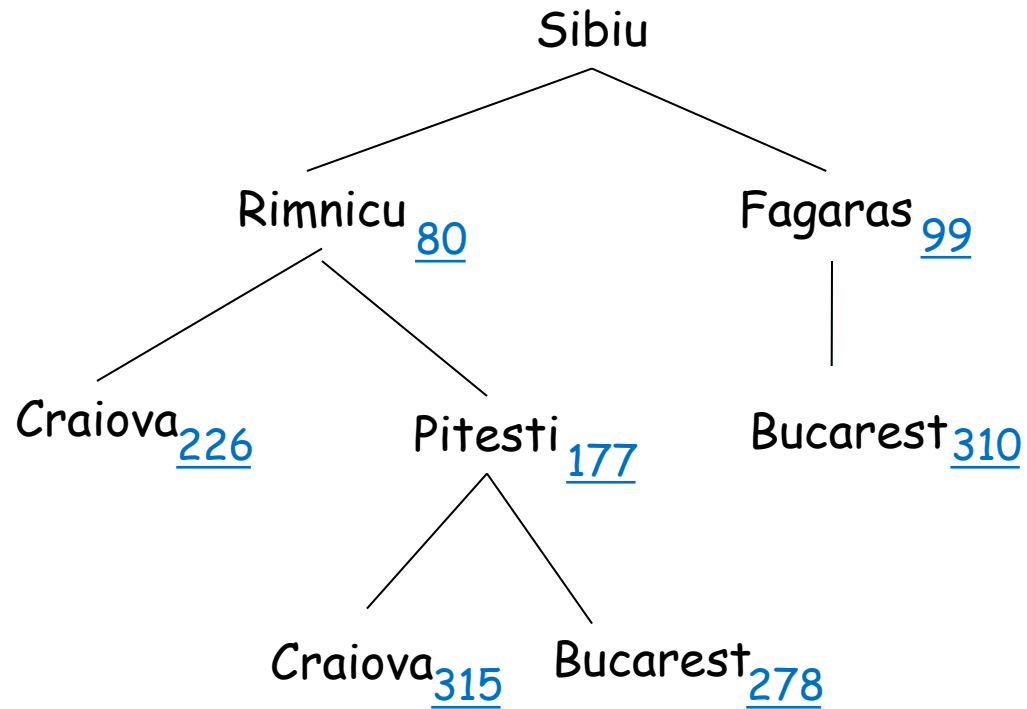
Búsqueda no informada



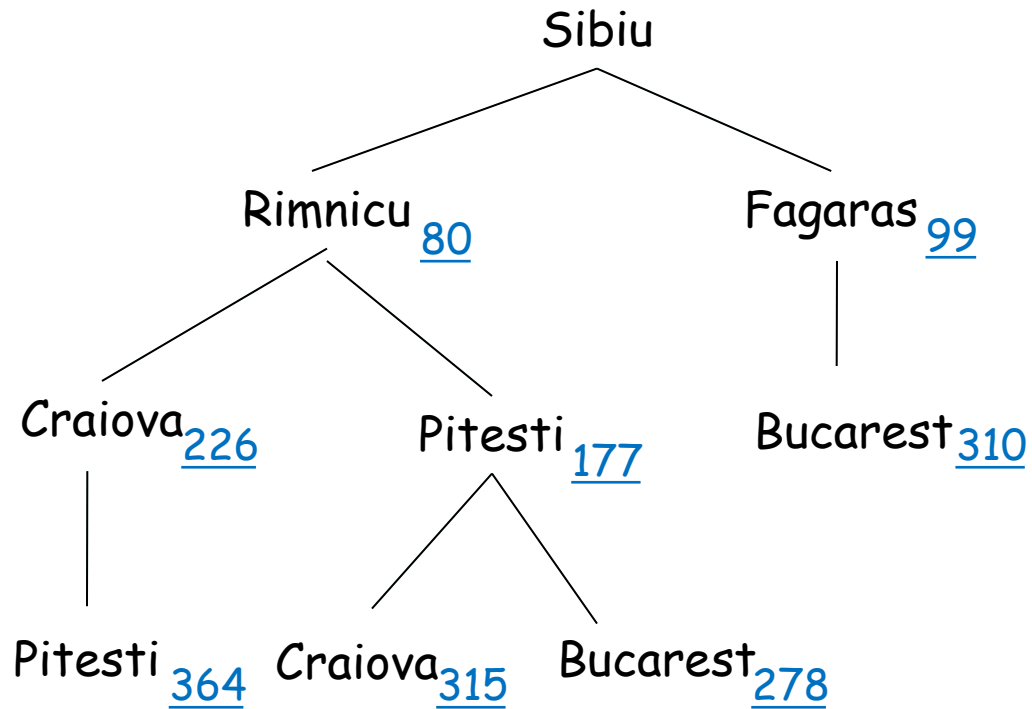
Búsqueda no informada



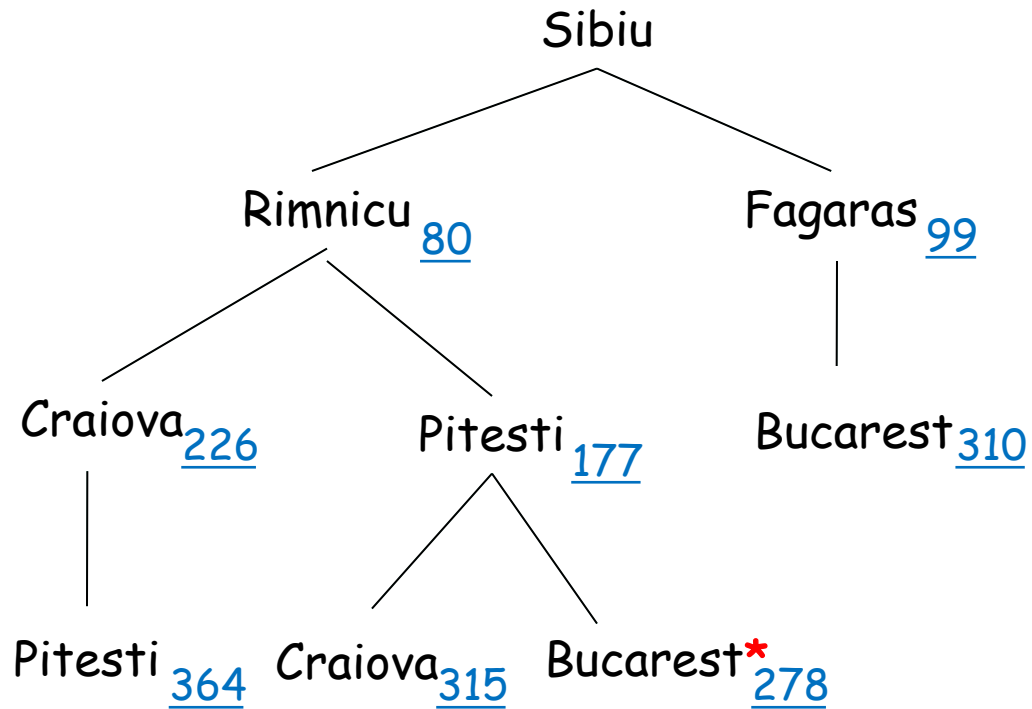
Búsqueda no informada



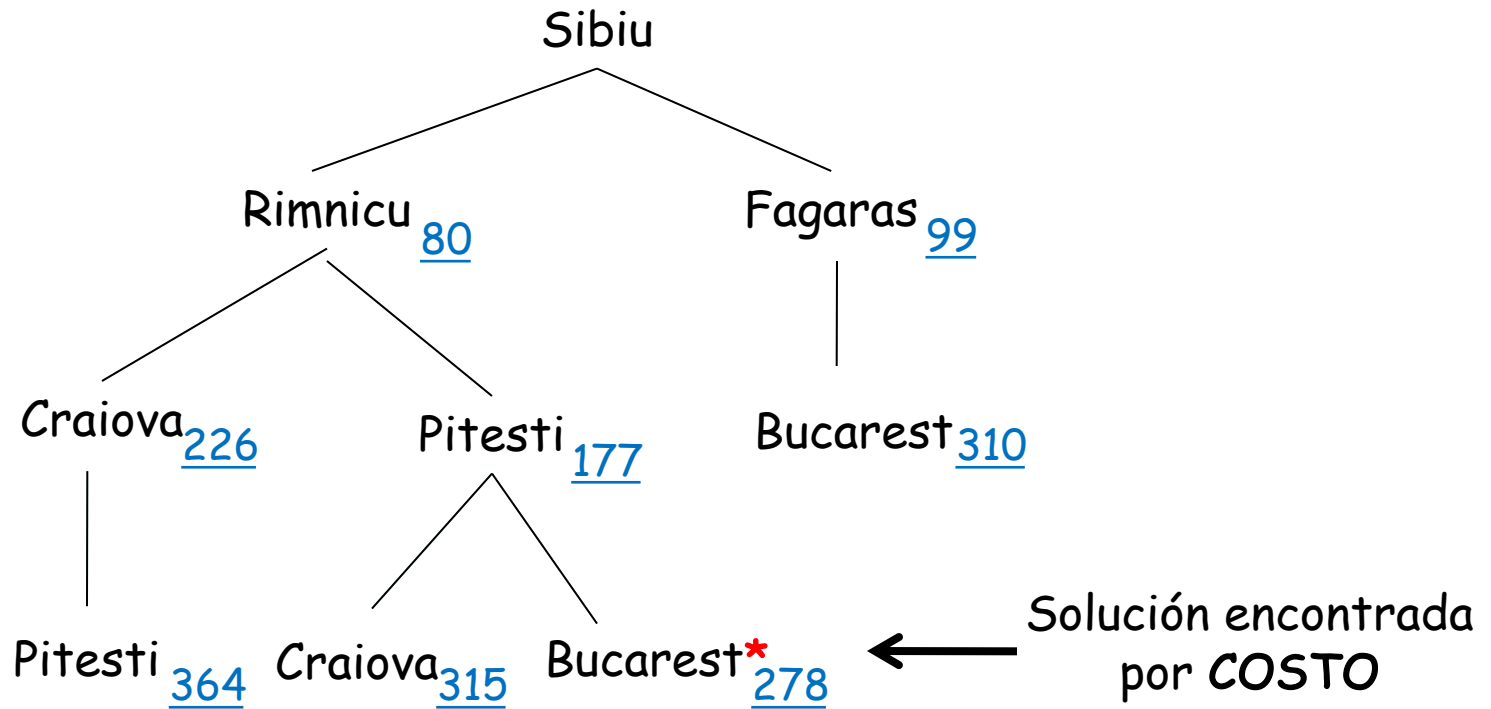
Búsqueda no informada



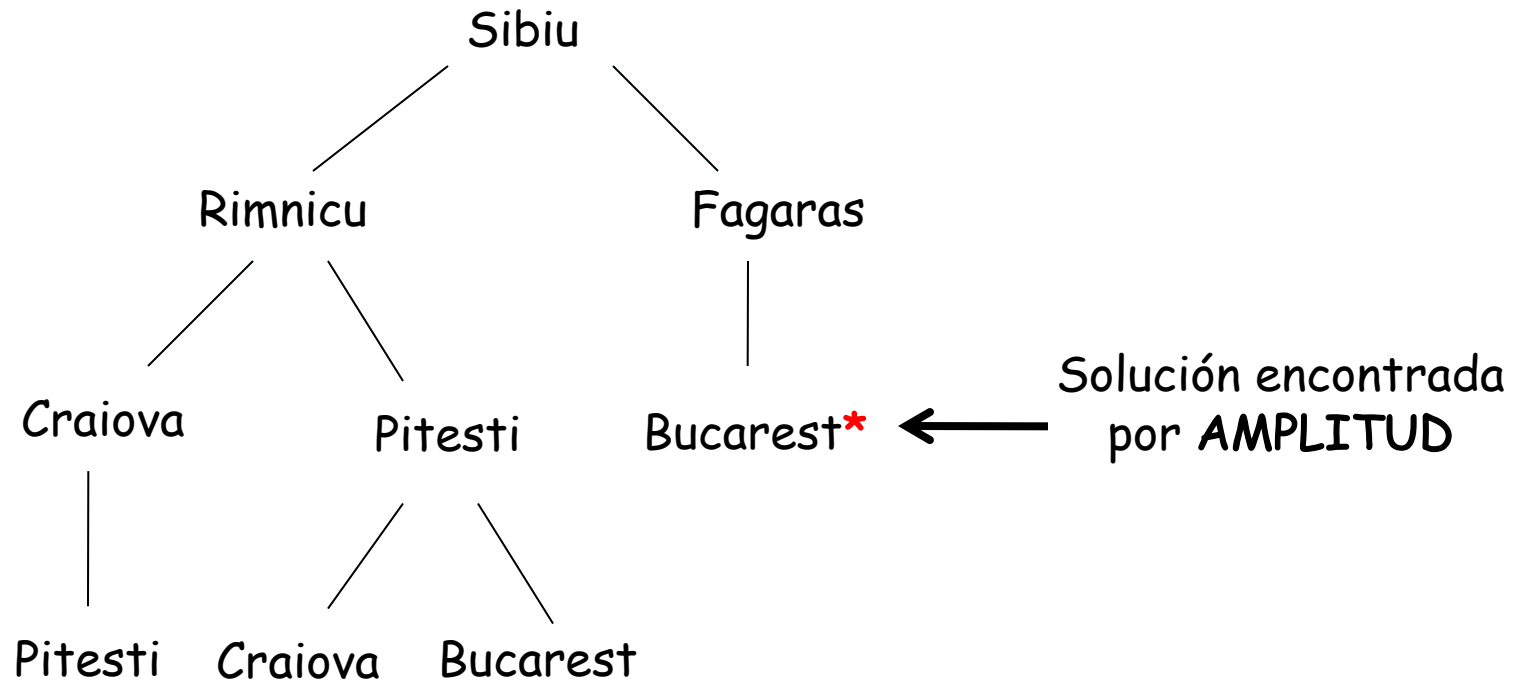
Búsqueda no informada



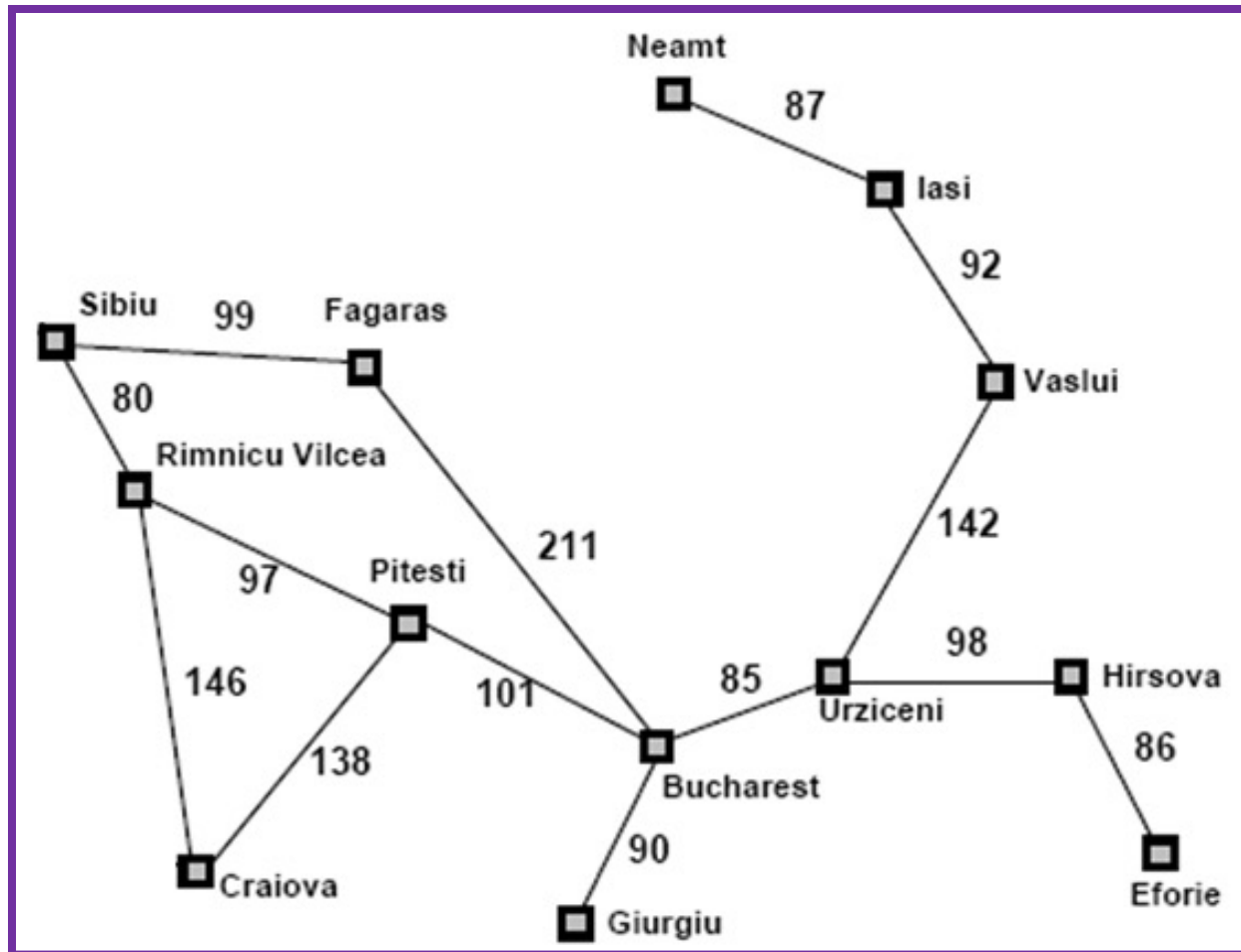
Búsqueda no informada



Búsqueda no informada



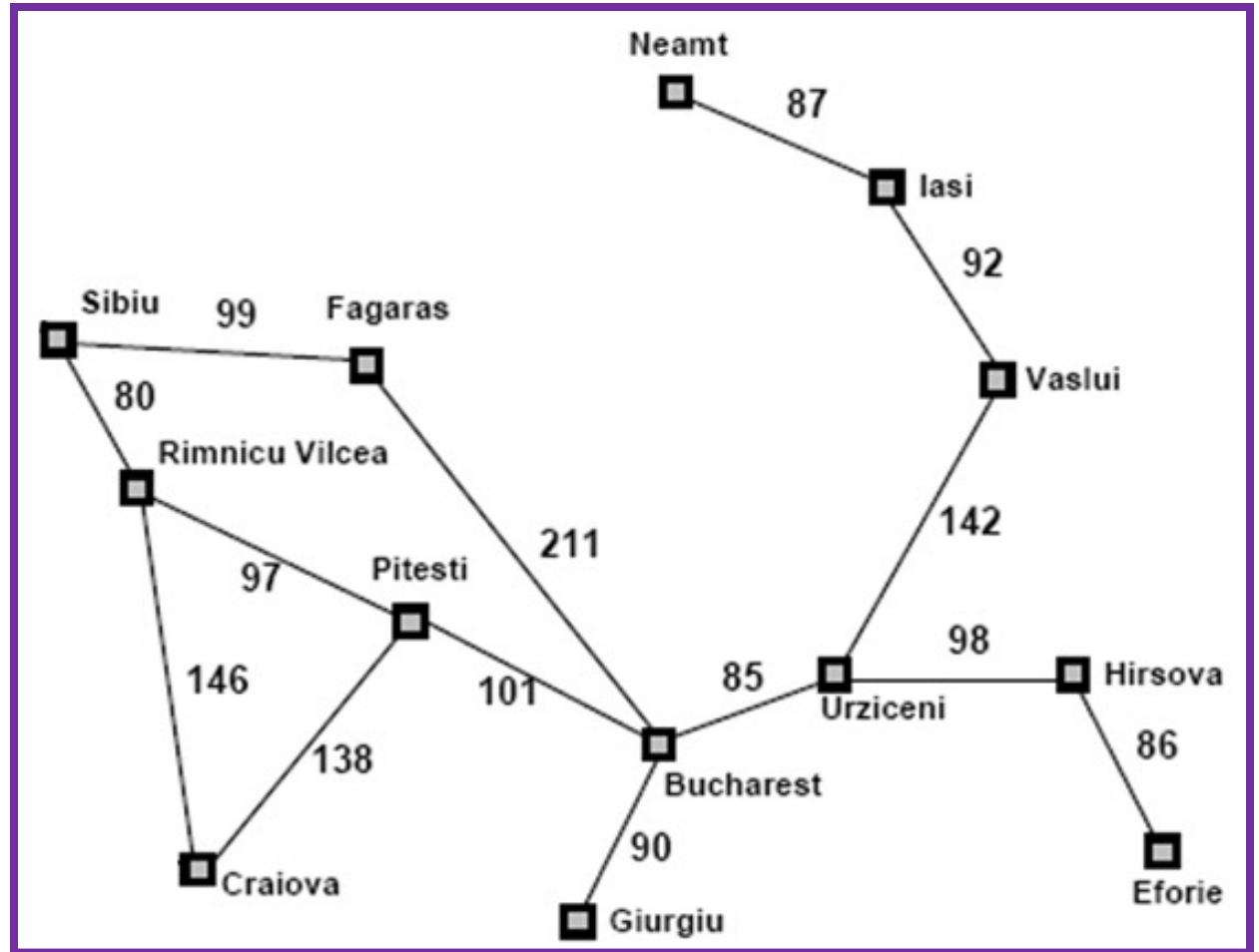
Búsqueda no informada



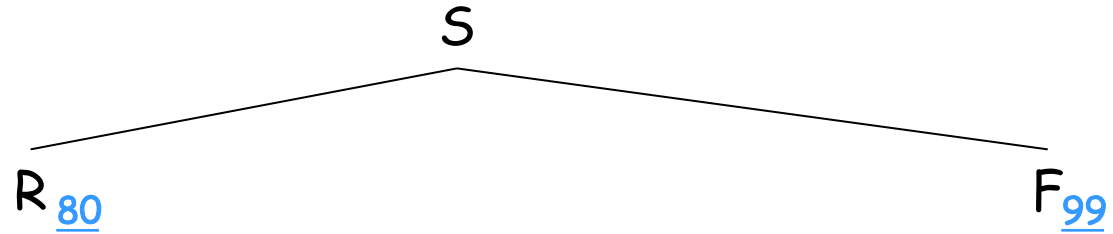
Búsqueda no informada

Aplicar búsqueda de costo uniforme
para ir de Sibiu a Bucarest

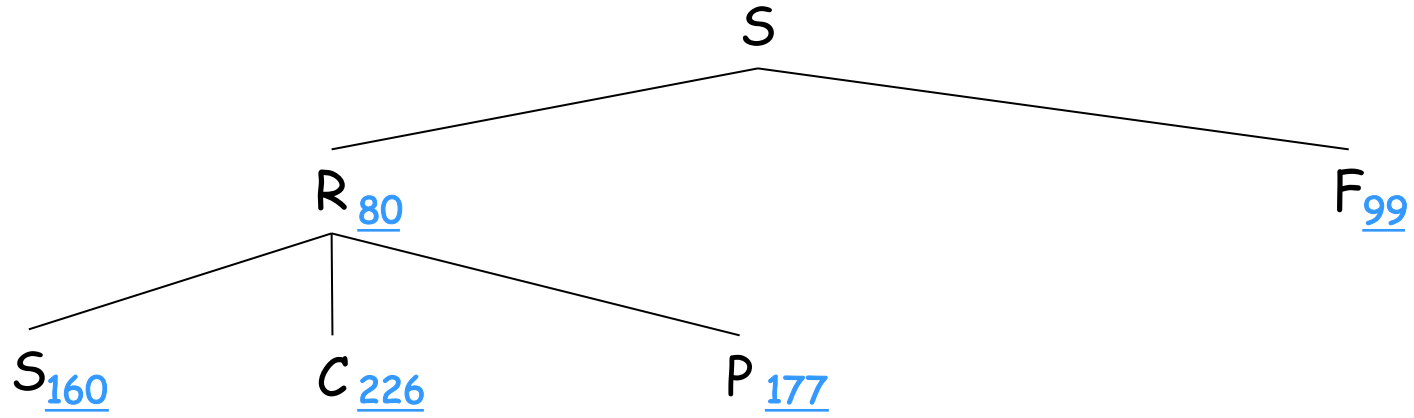
No evite devolverse



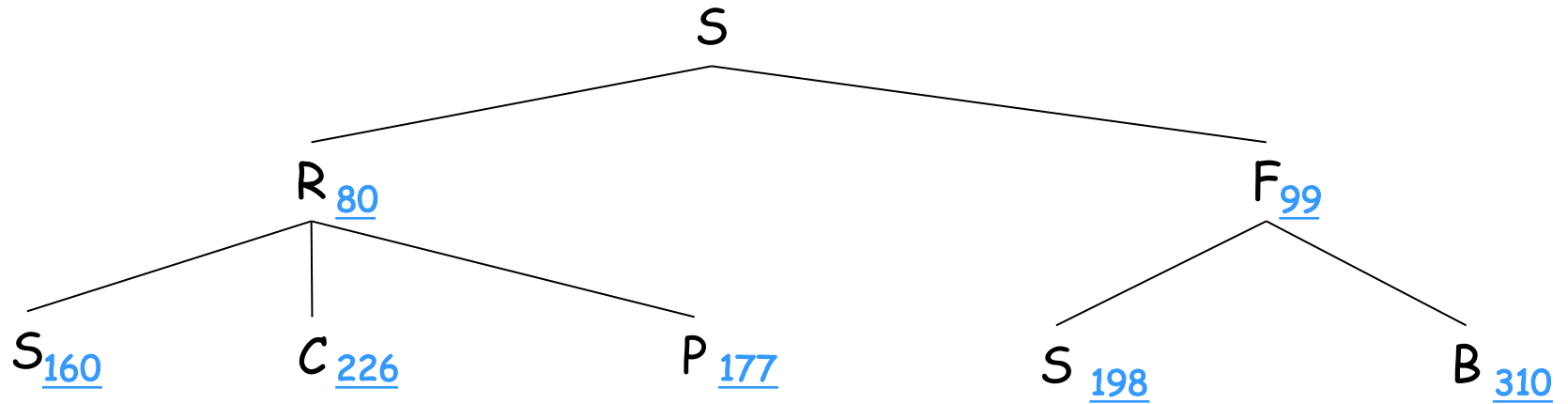
Búsqueda no informada



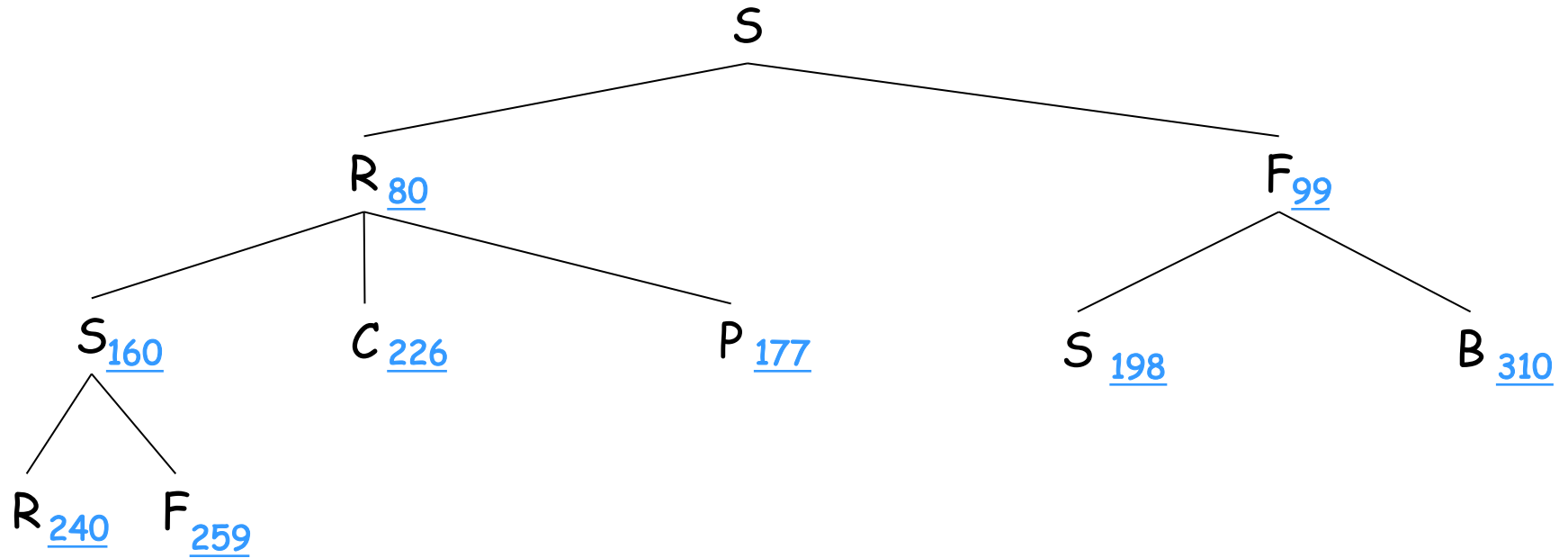
Búsqueda no informada



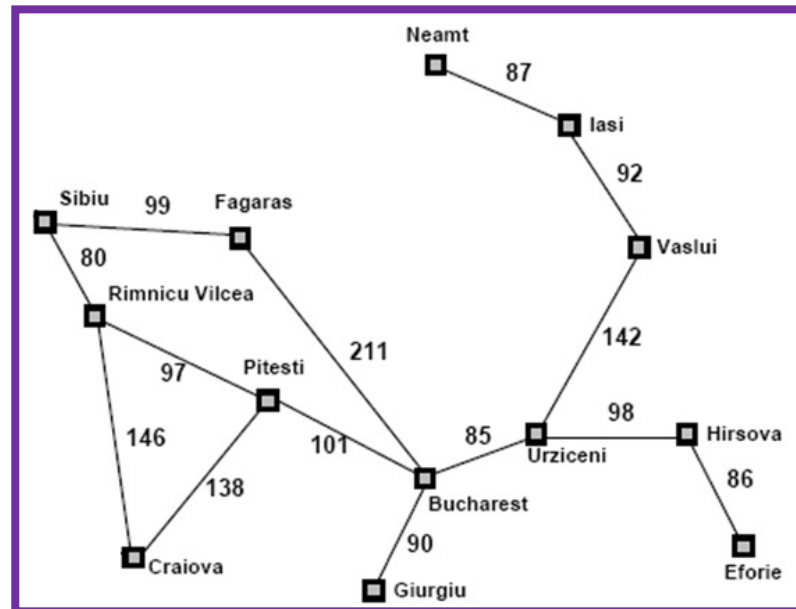
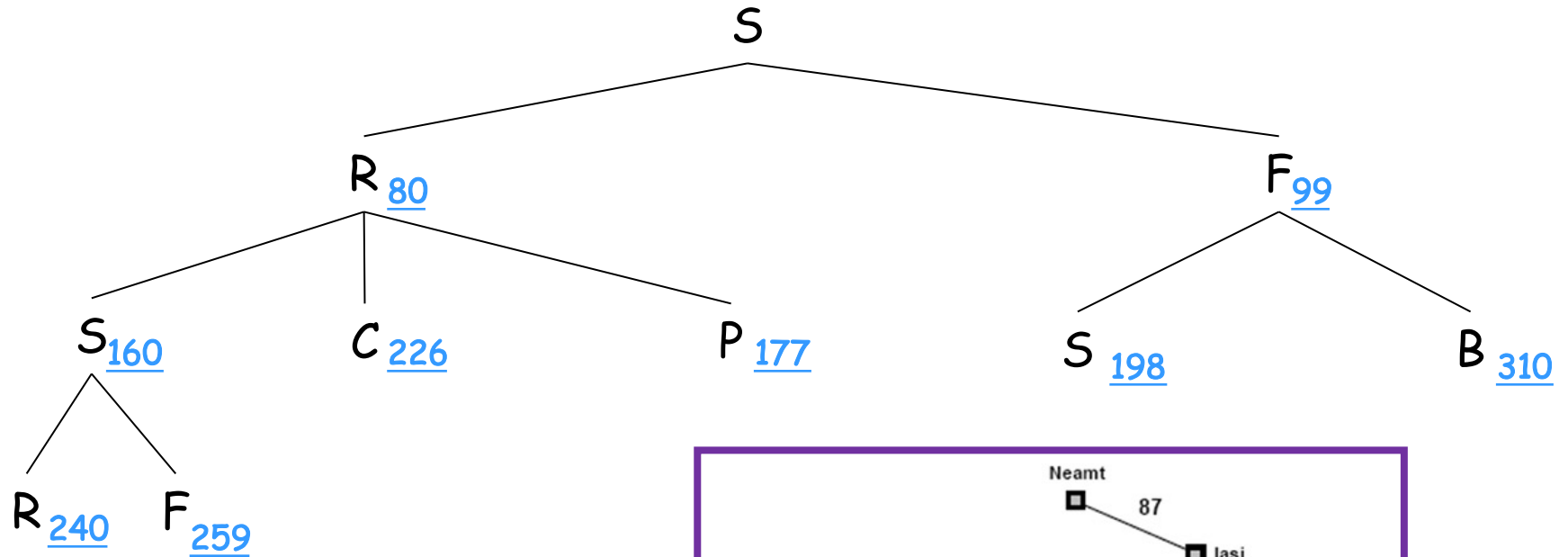
Búsqueda no informada



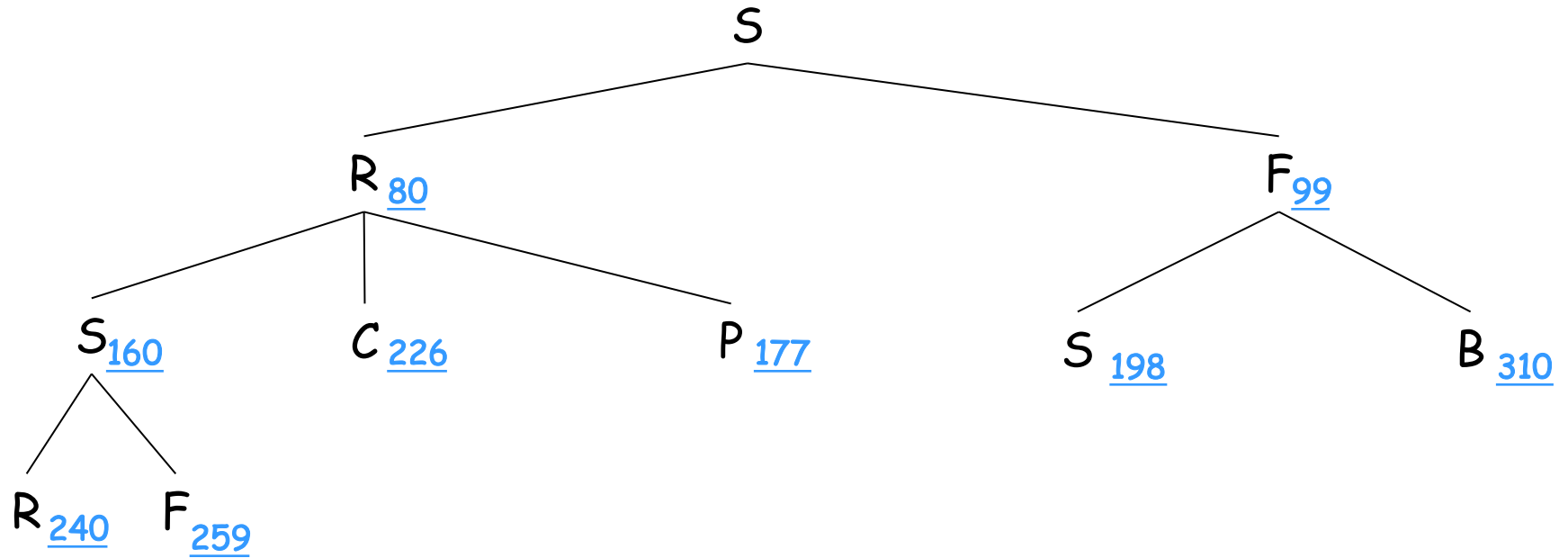
Búsqueda no informada



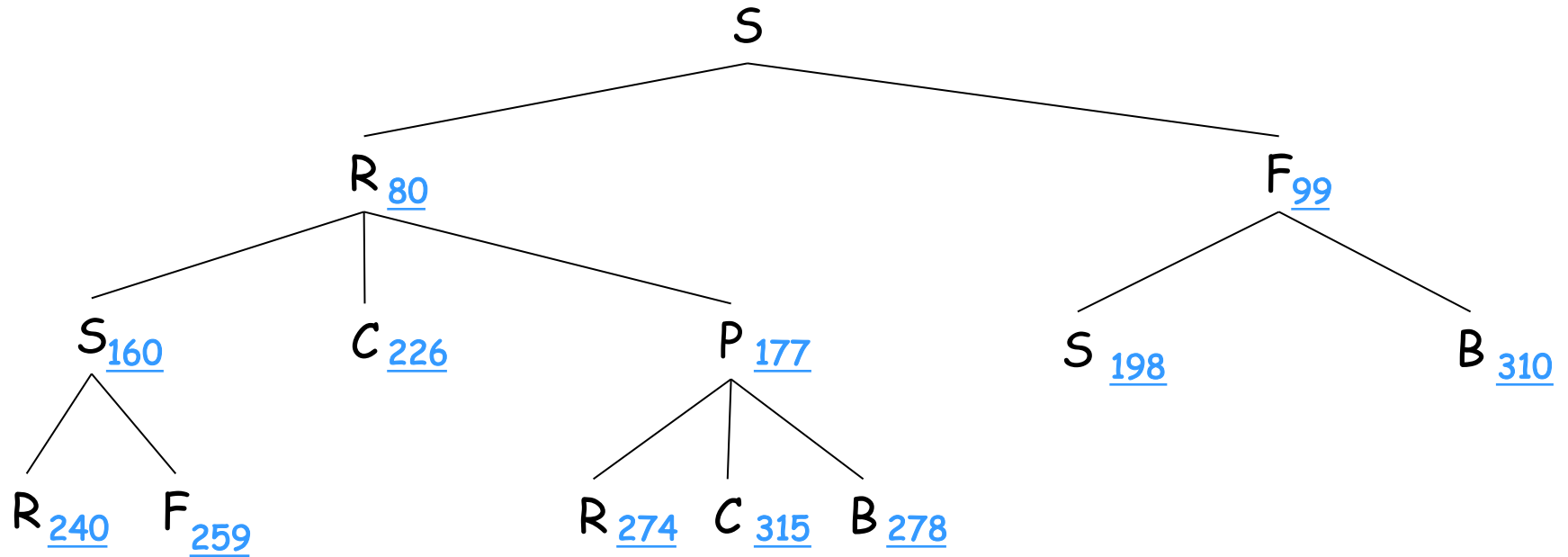
Búsqueda no informada



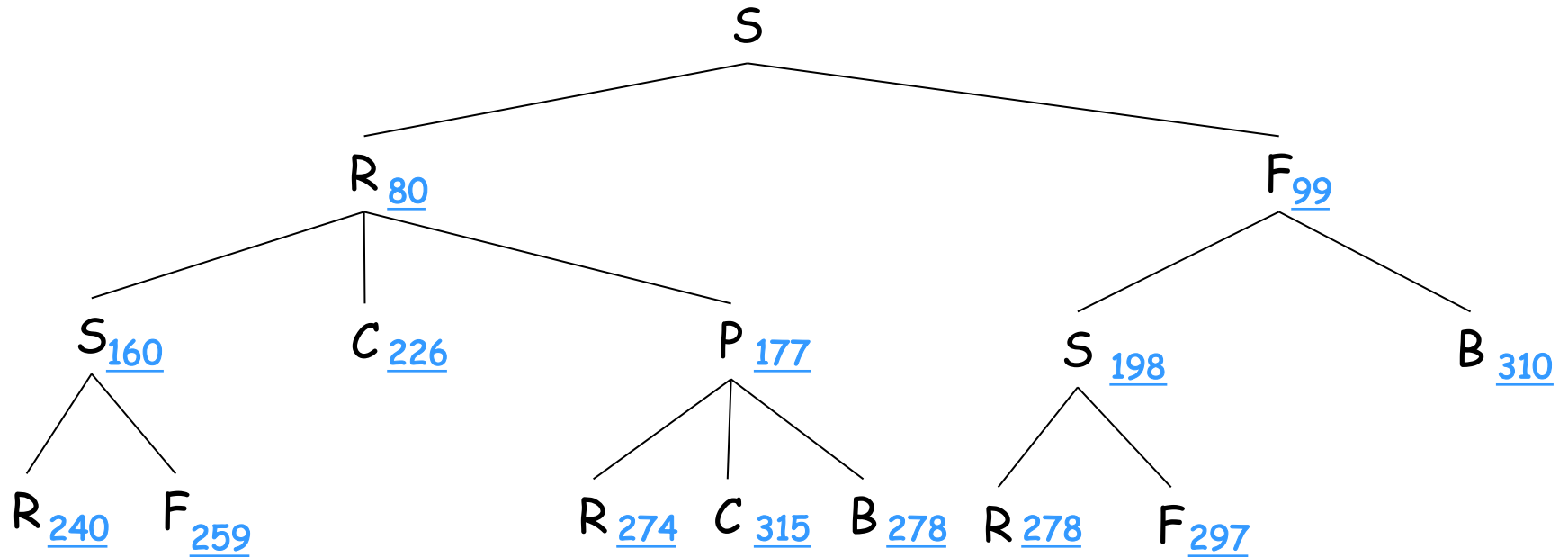
Búsqueda no informada



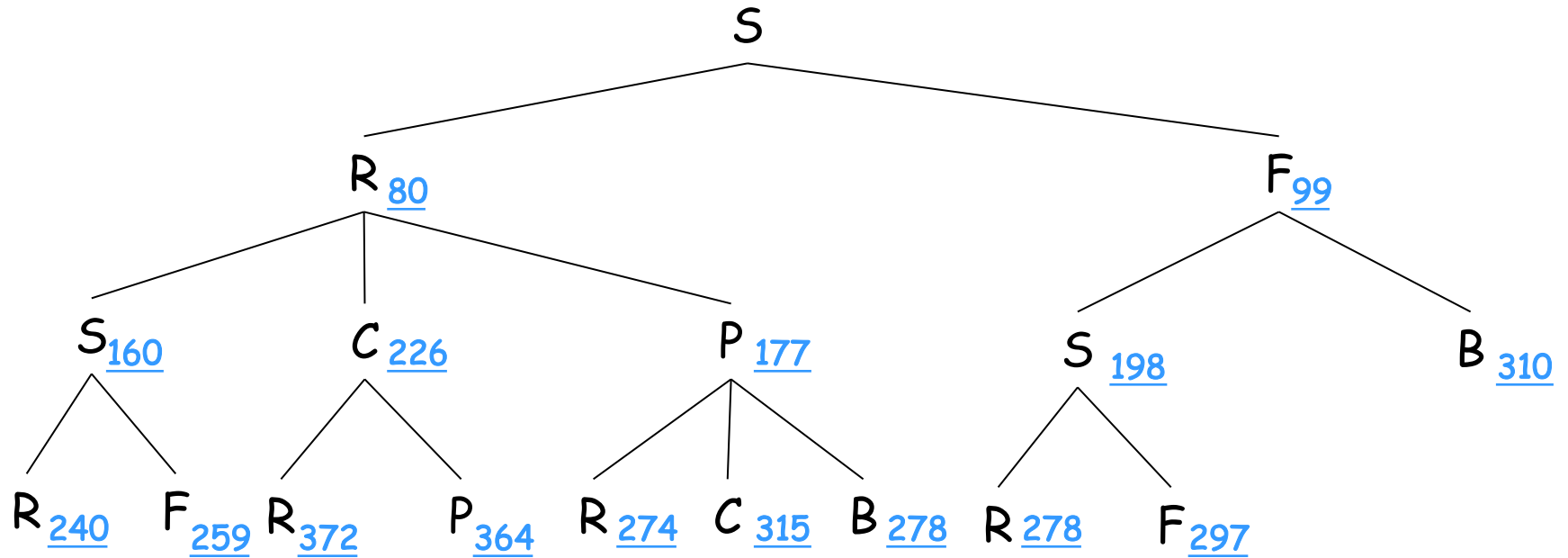
Búsqueda no informada



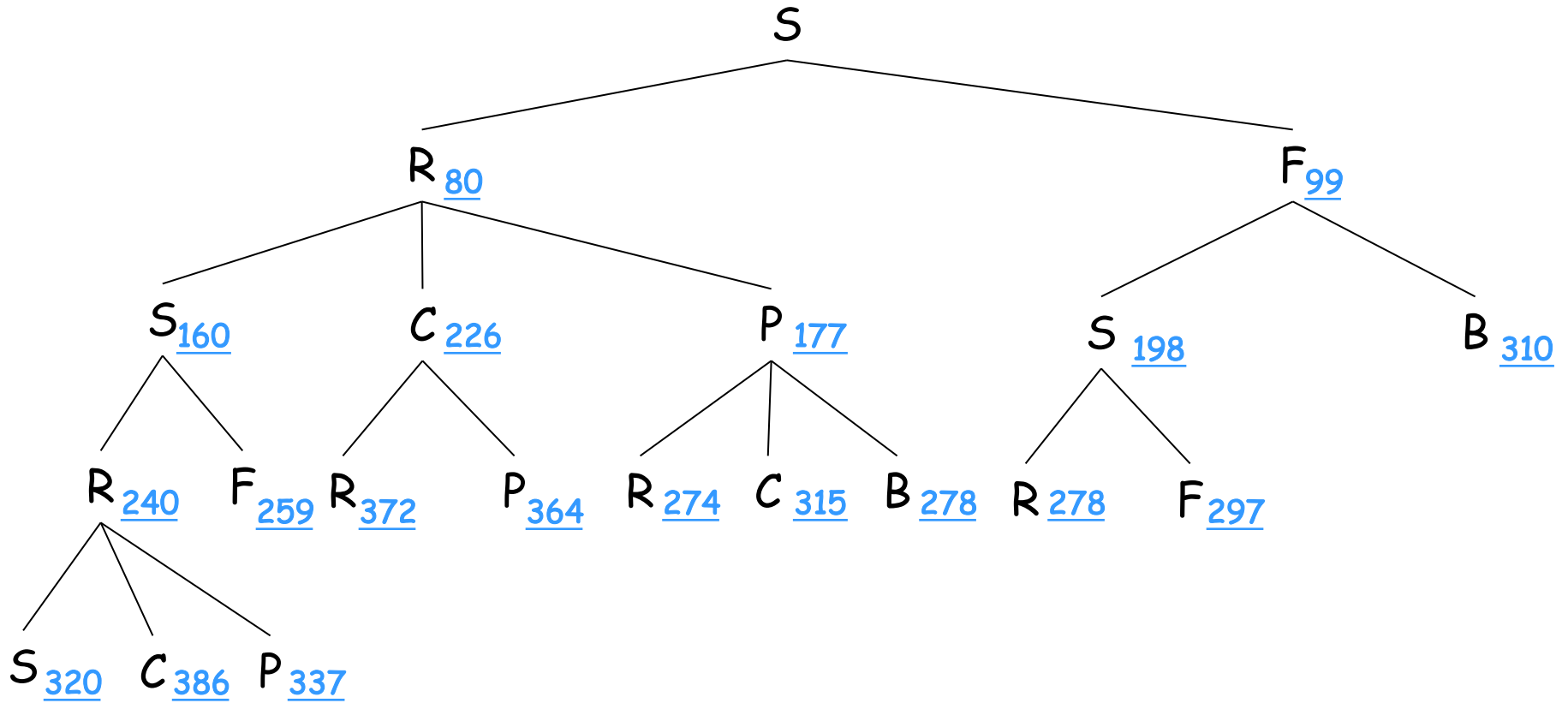
Búsqueda no informada



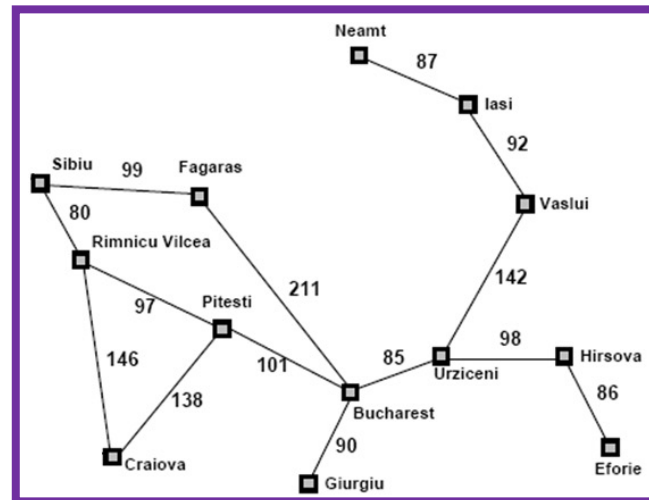
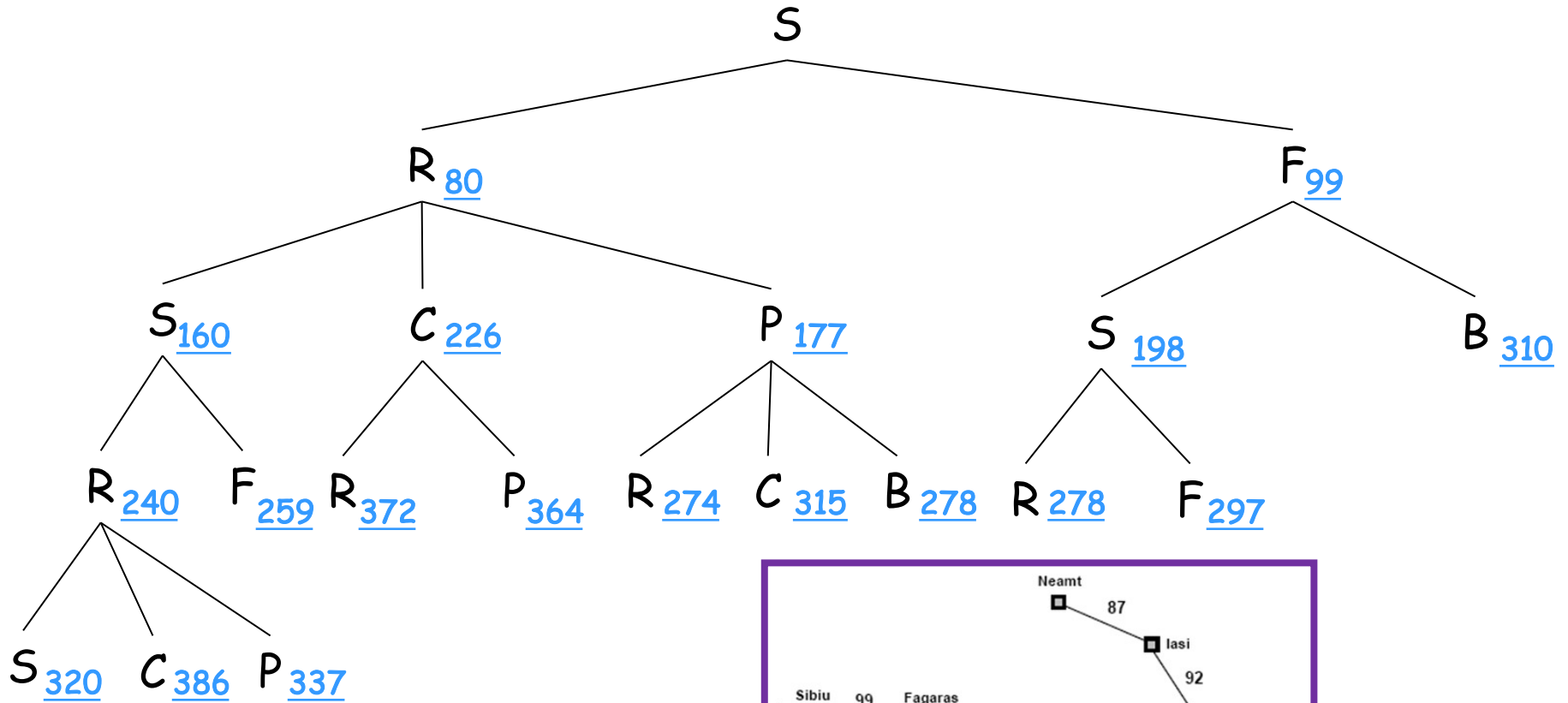
Búsqueda no informada



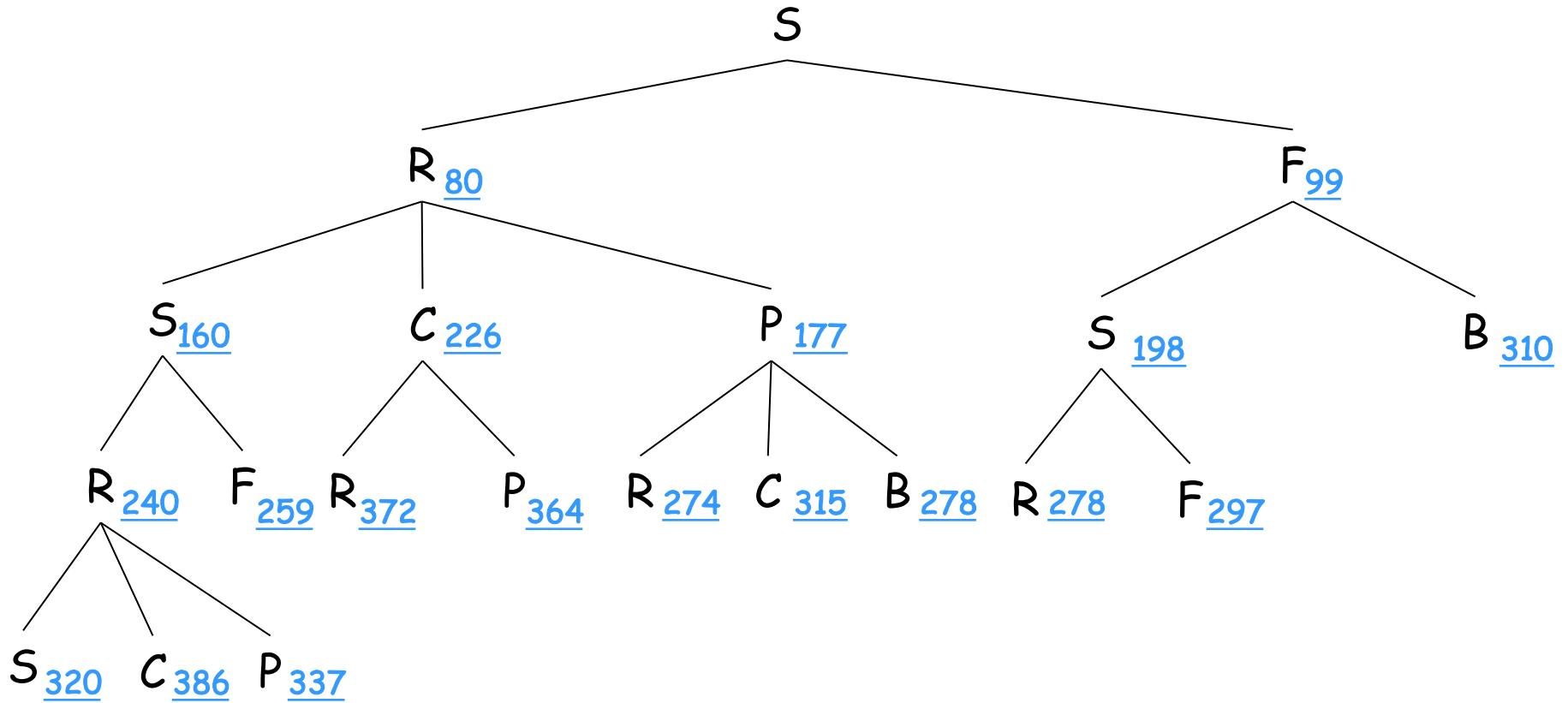
Búsqueda no informada



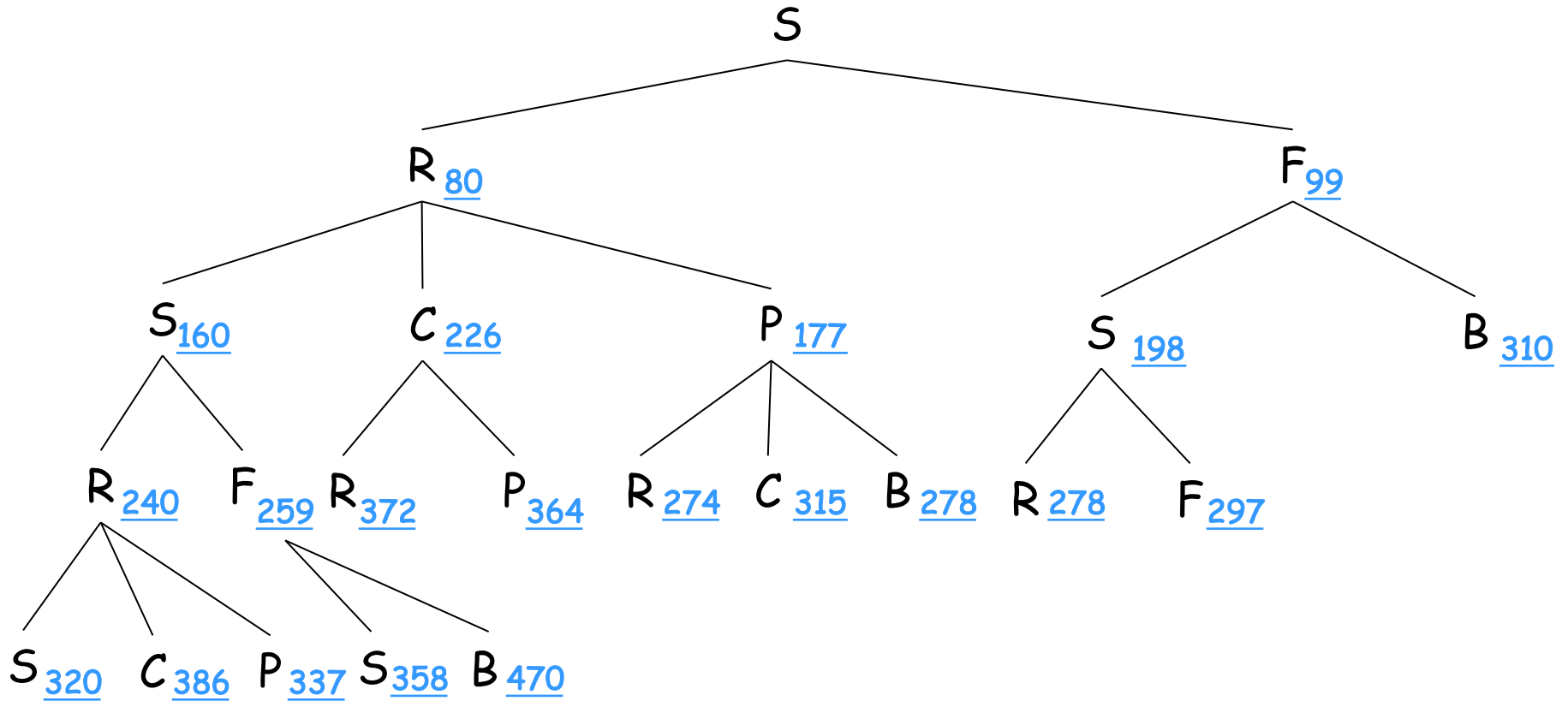
Búsqueda no informada



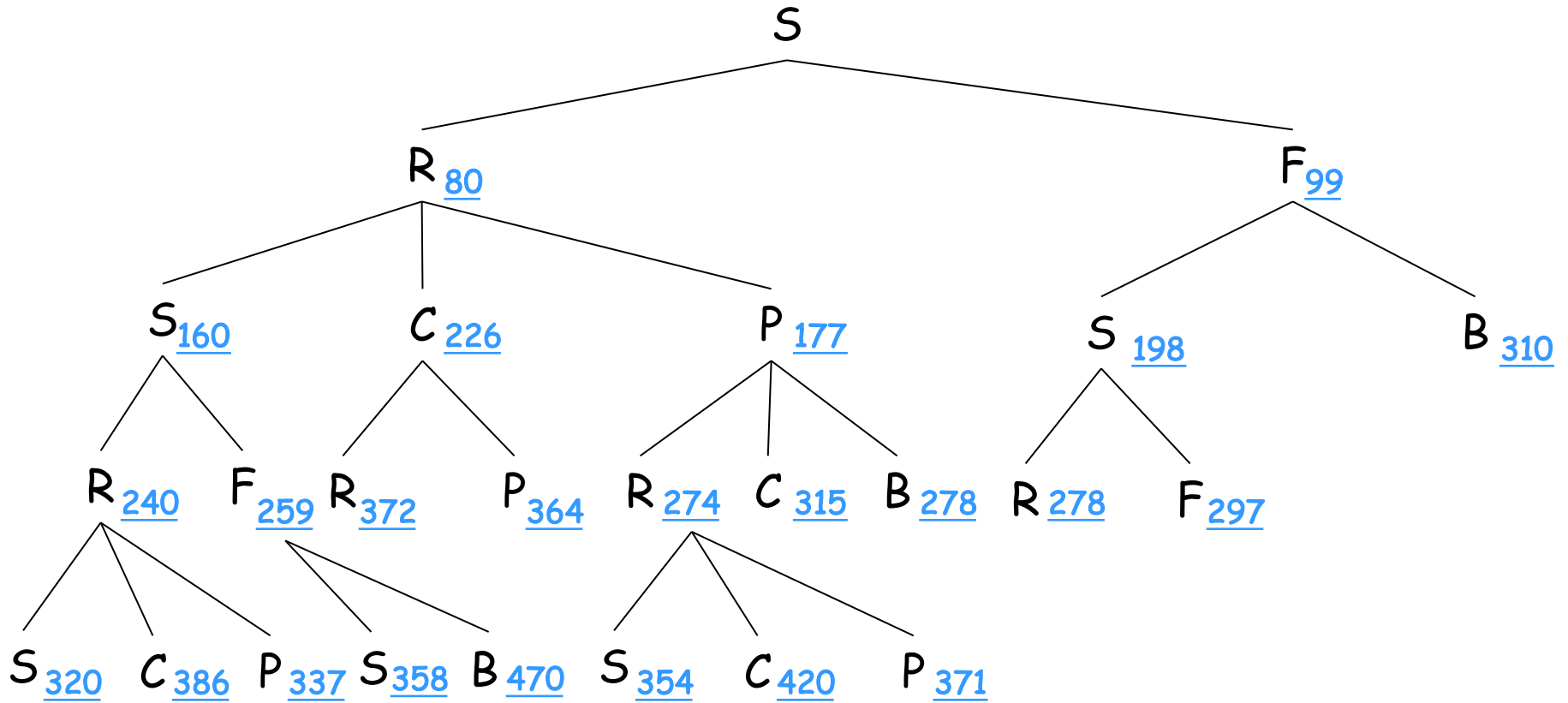
Búsqueda no informada



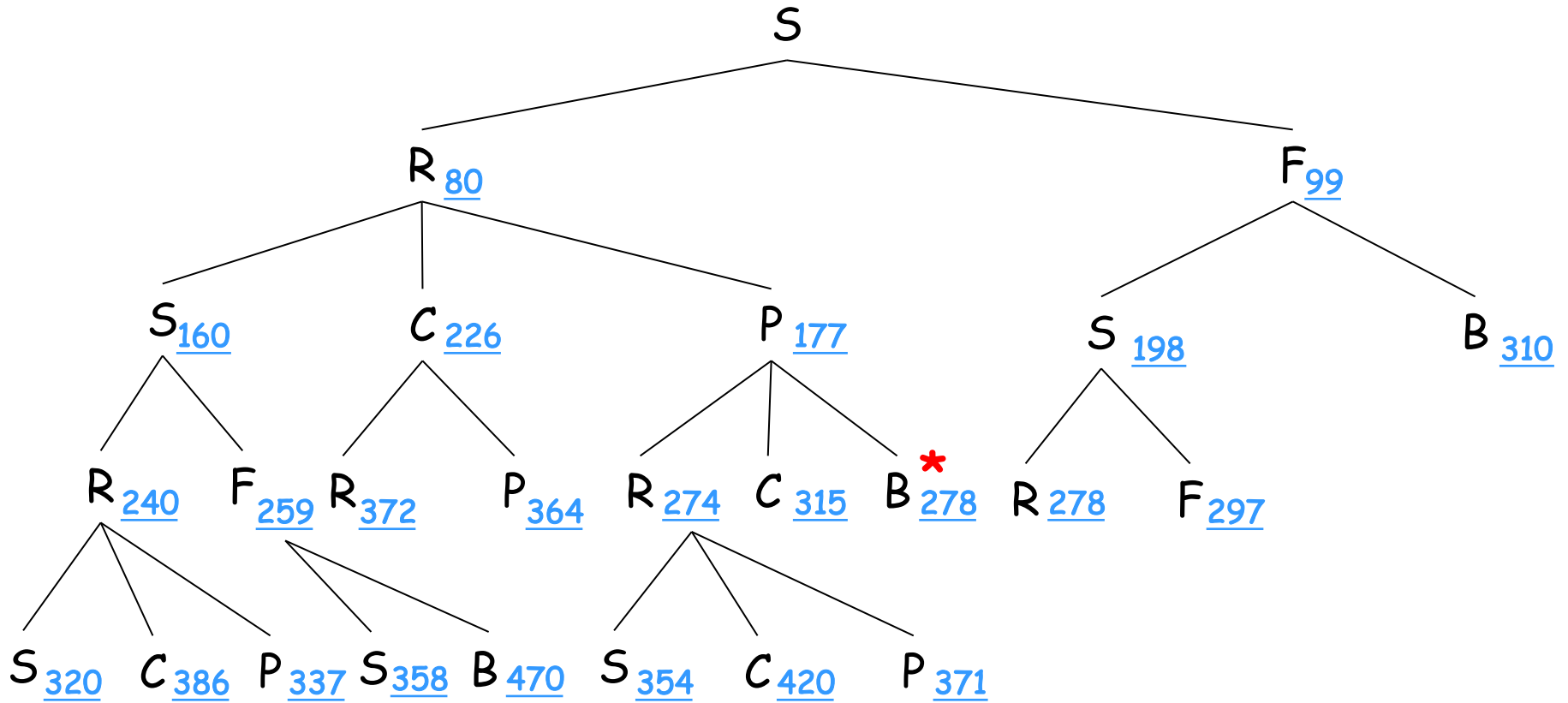
Búsqueda no informada



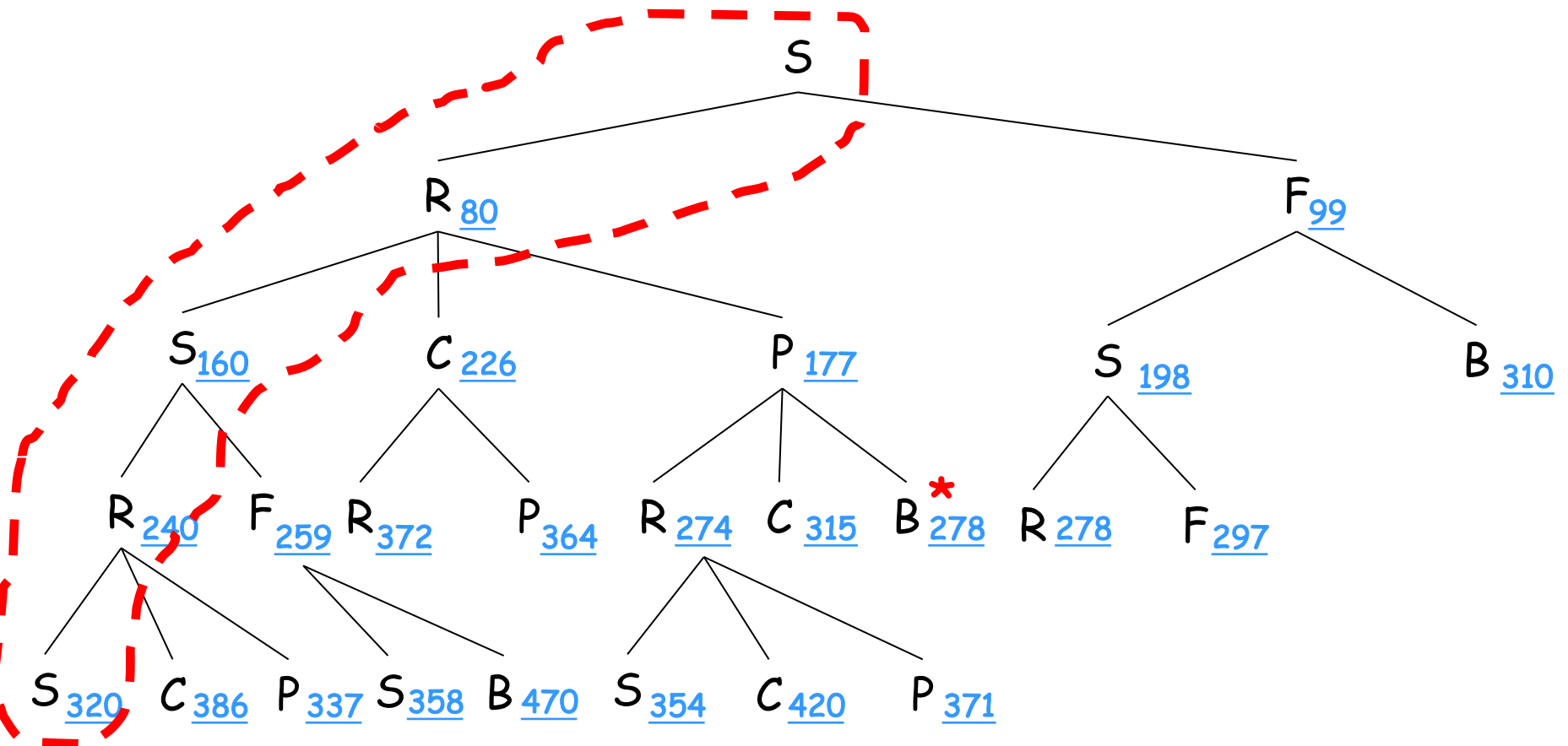
Búsqueda no informada



Búsqueda no informada



Búsqueda no informada

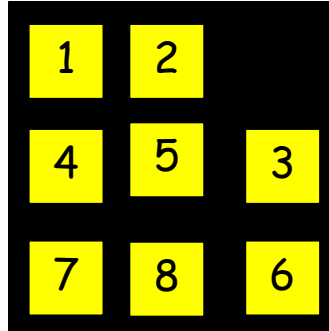


Los ciclos no afectan la búsqueda por costo uniforme porque al repetir estados se acumula el costo de ruta

Búsqueda no informada

Aplicar búsqueda de
costo uniforme

Evite devolverse



Cada movimiento cuesta 1

1	2	-
4	5	3
7	8	6

1	2	-
4	5	3
7	8	6

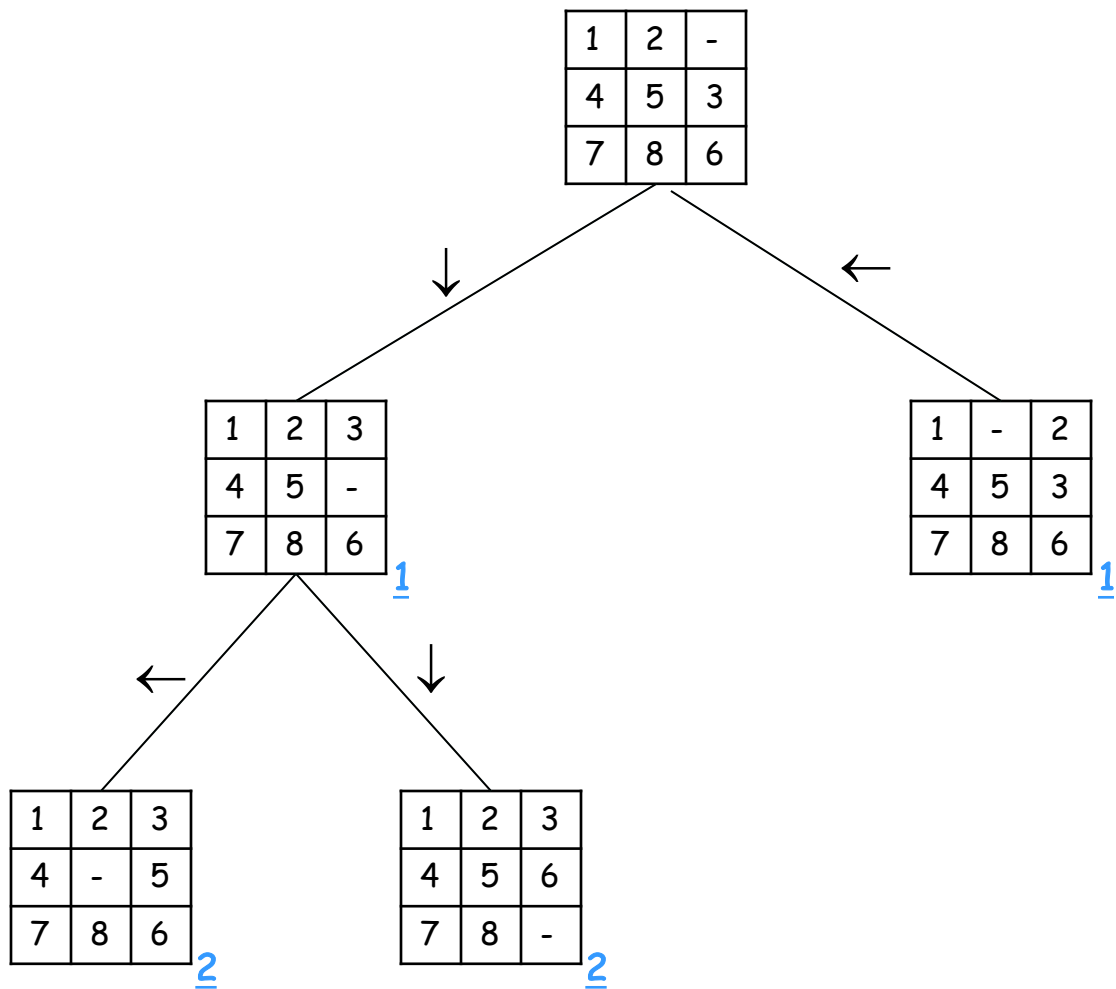


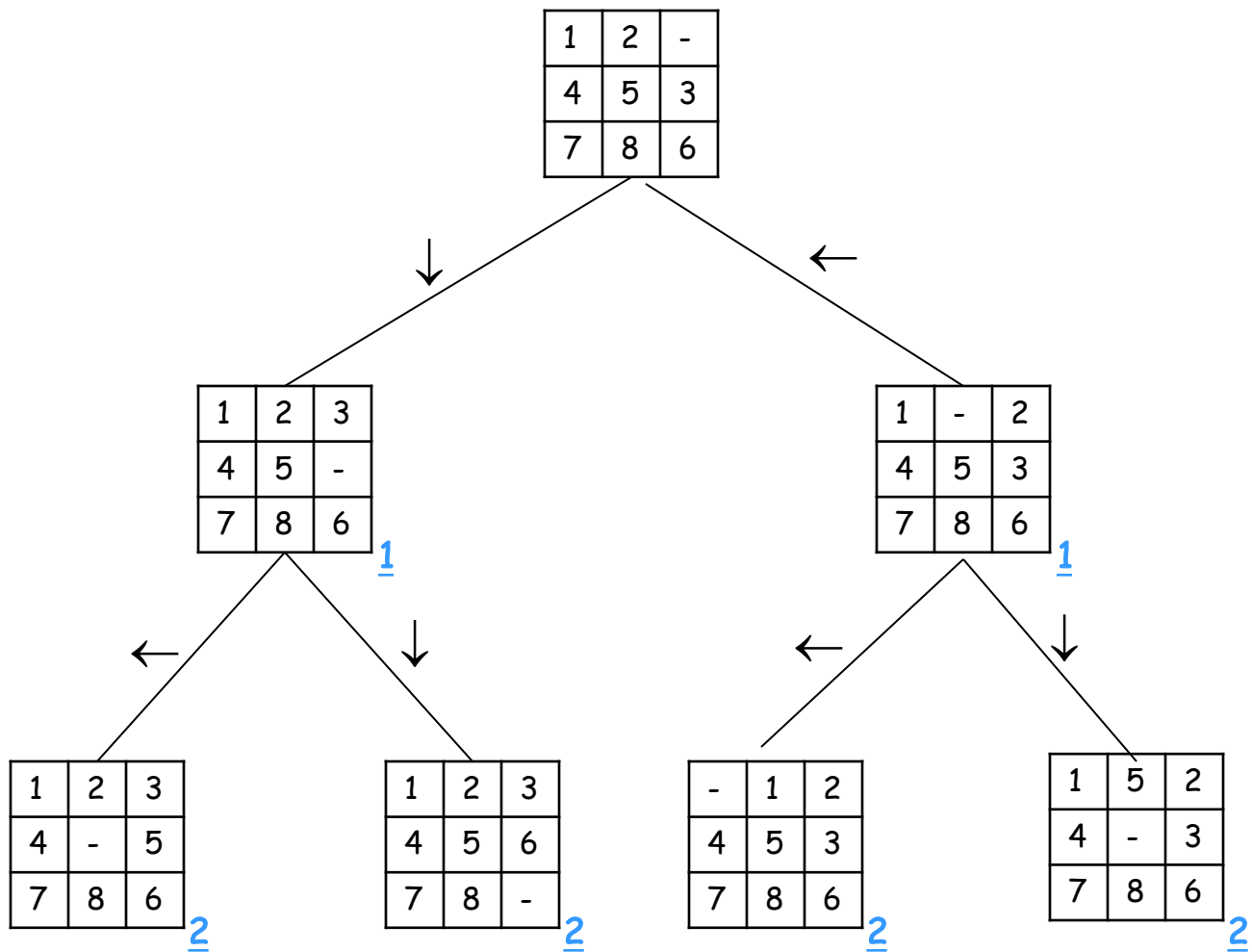
1	2	3
4	5	-
7	8	6

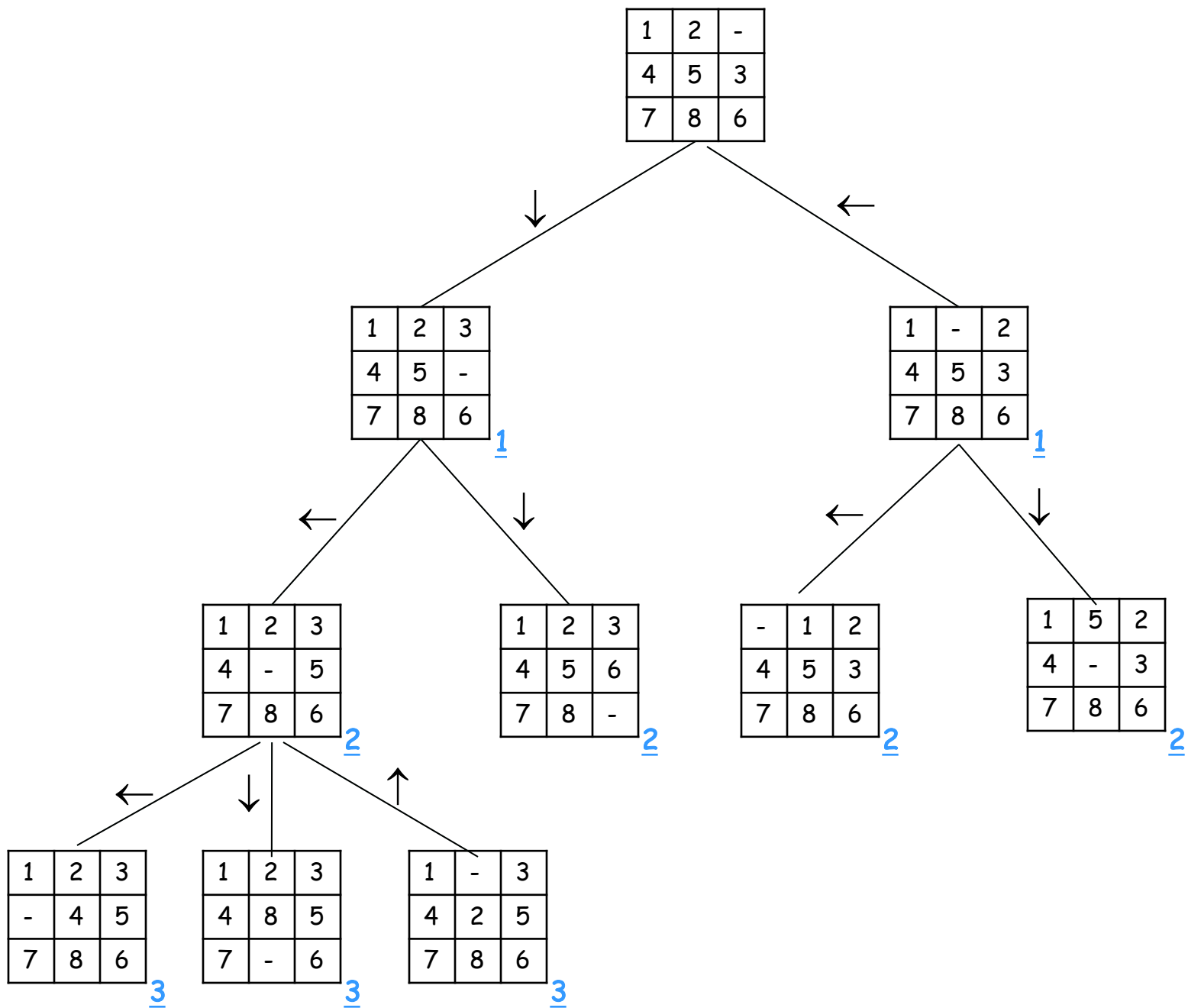
1

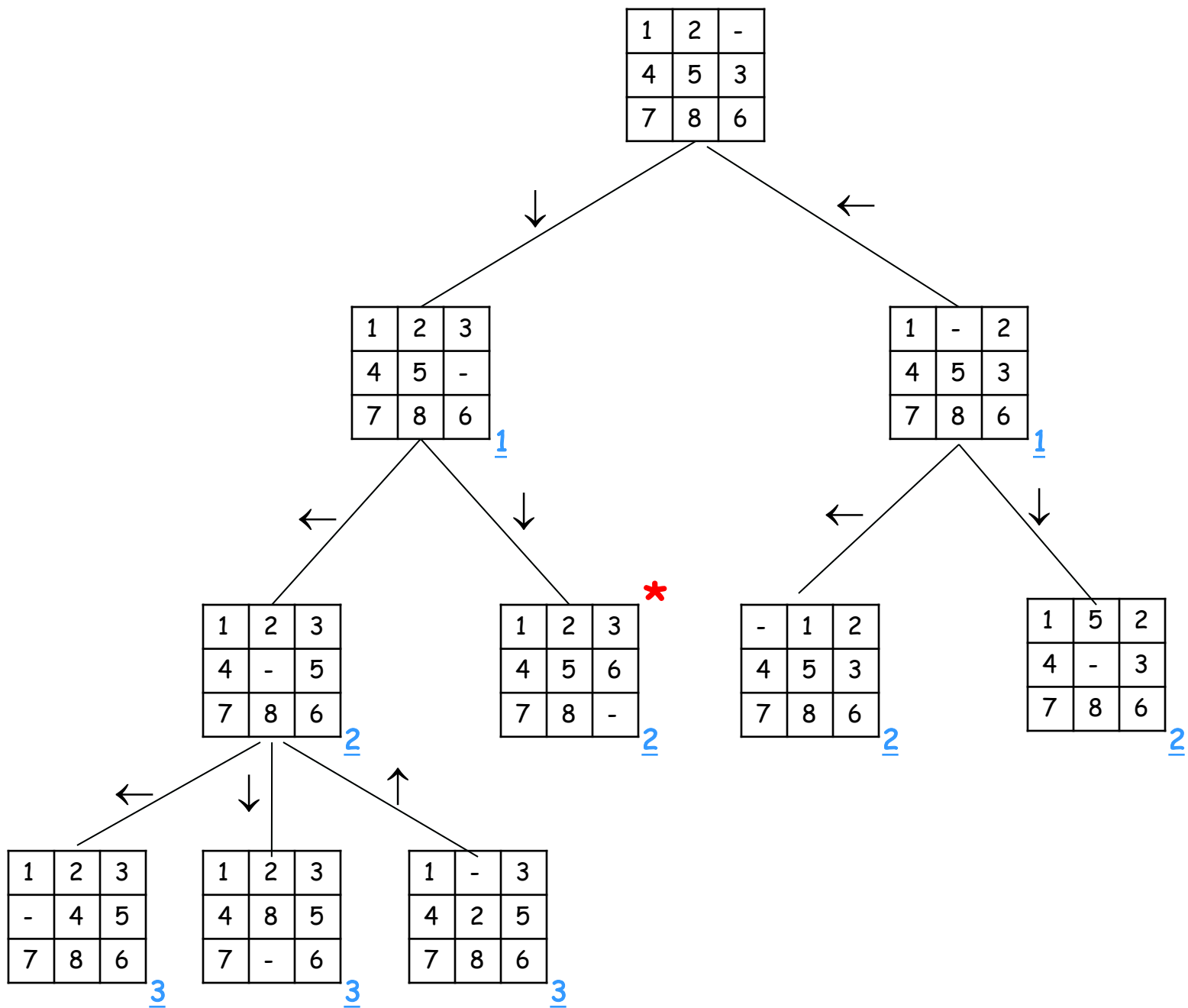
1	-	2
4	5	3
7	8	6

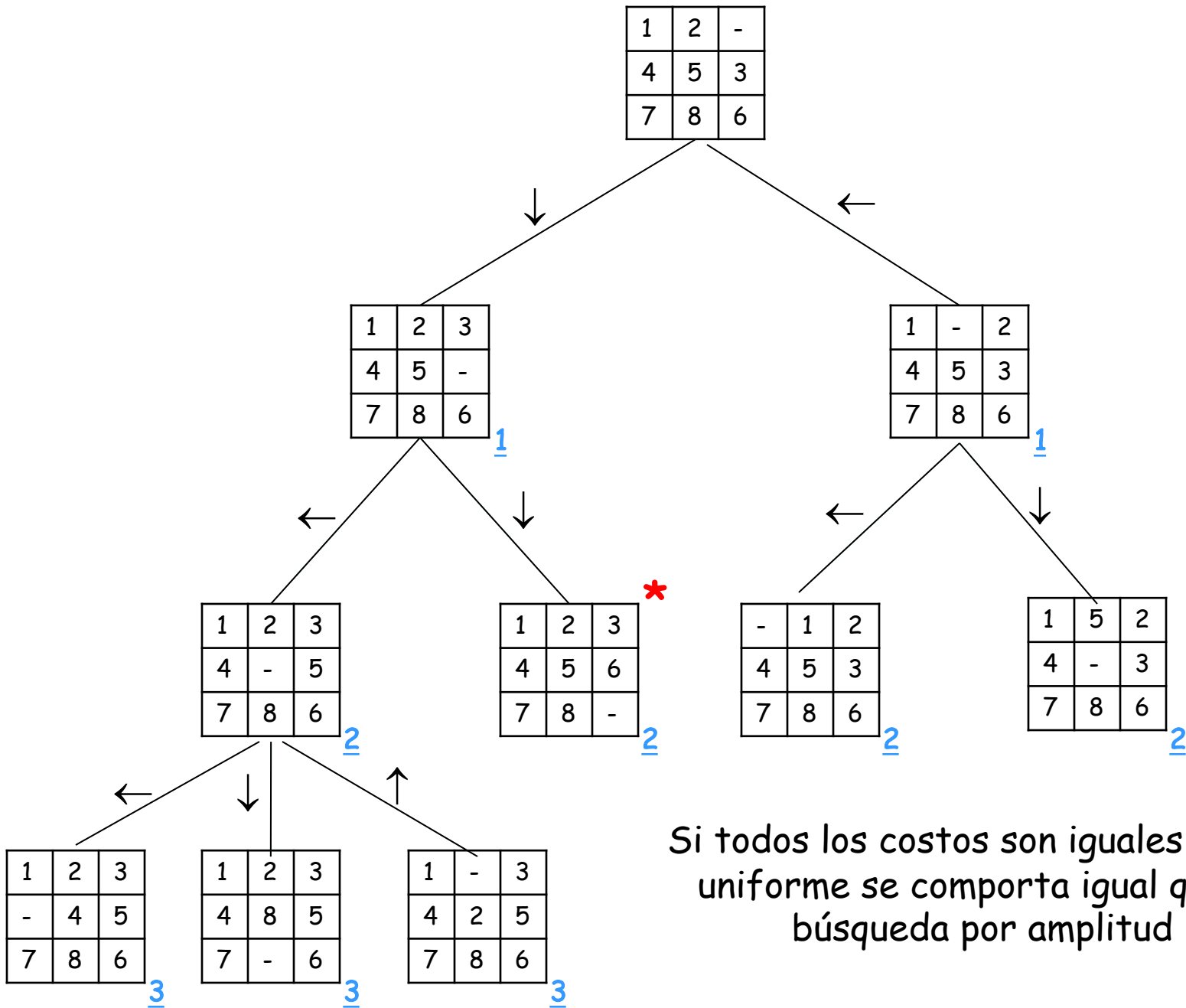
1











Si todos los costos son iguales, costo uniforme se comporta igual que la búsqueda por amplitud





Búsqueda no informada


Aplicar búsqueda de costo uniforme

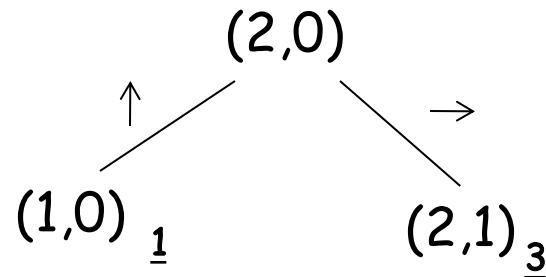
Operadores

- Arriba
- Abajo
- Izquierda
- Derecha

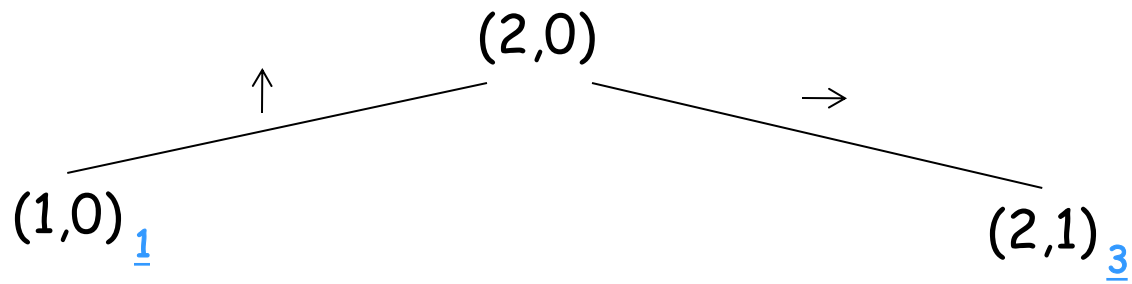
Los costos varían de acuerdo al color de la casilla a la cual se llegue

-  : 1
-  : 2
-  : 3
-  : 1

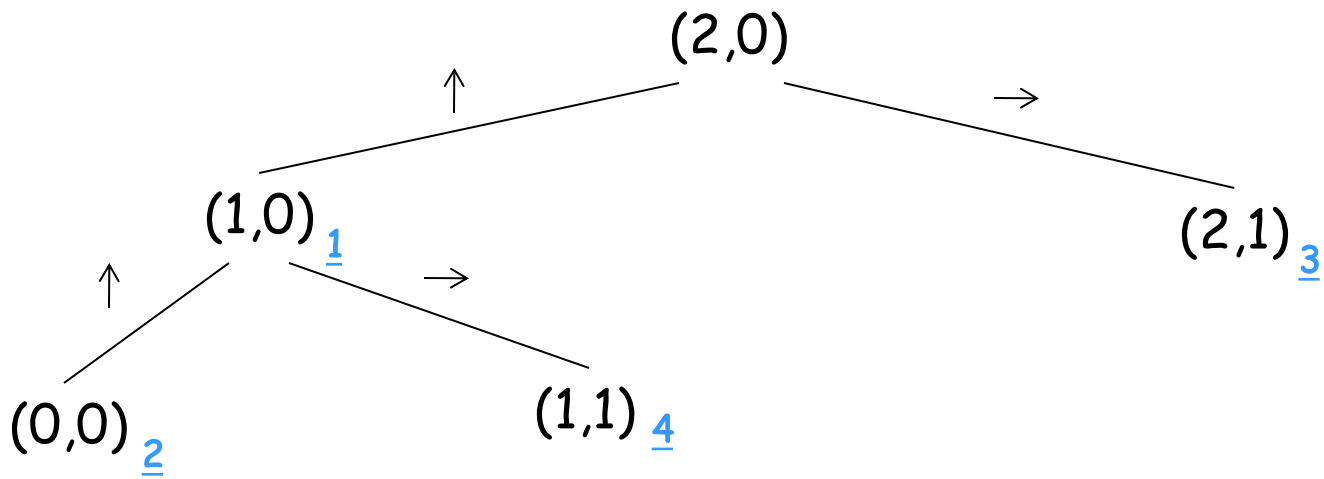
	0	1	2	3
0				
1				 Fin
2	 Inicio			



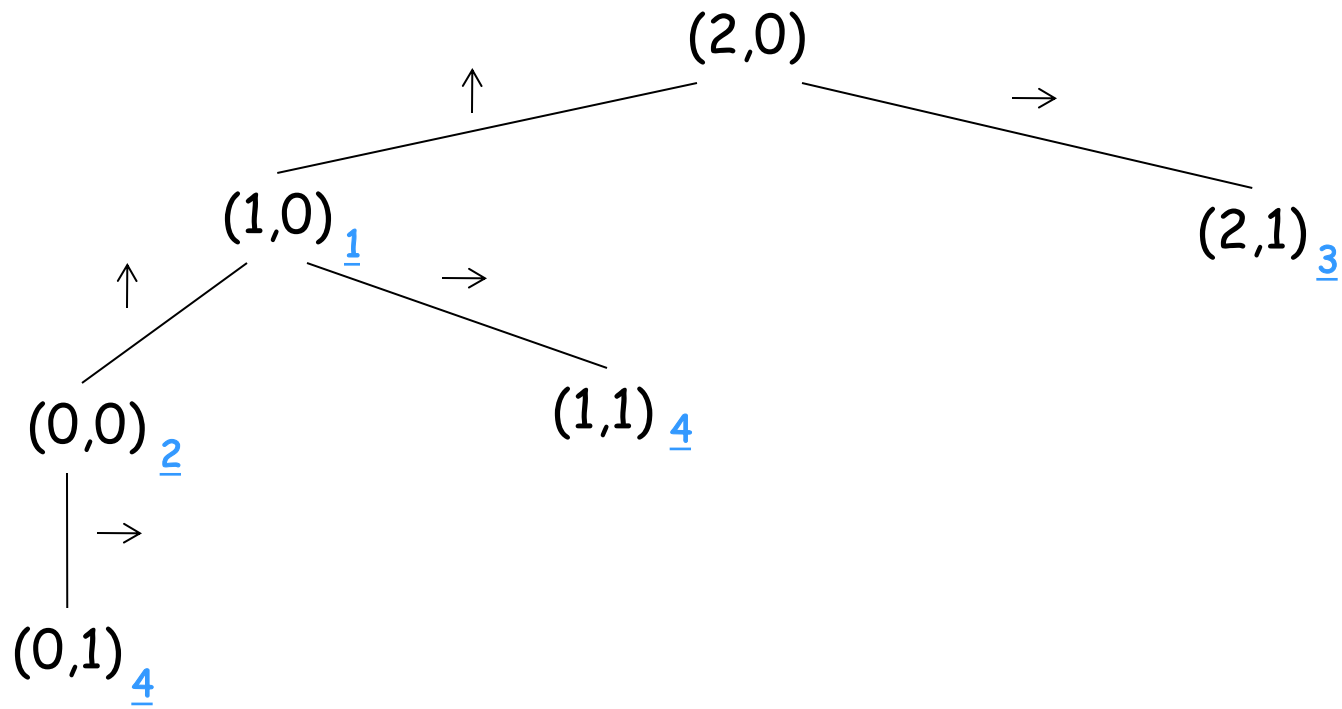
* Evite devolverse



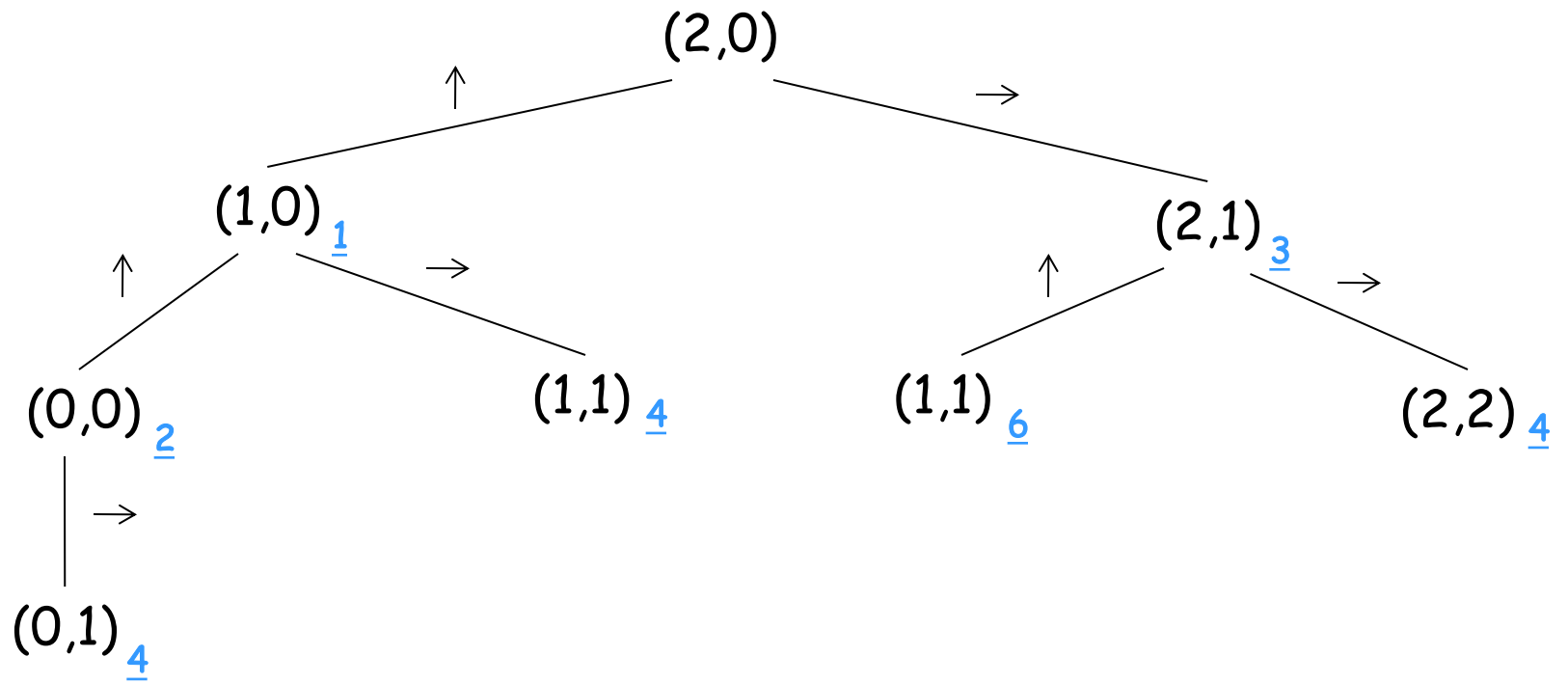
	0	1	2	3
0				
1				Fin
2	Inicio			



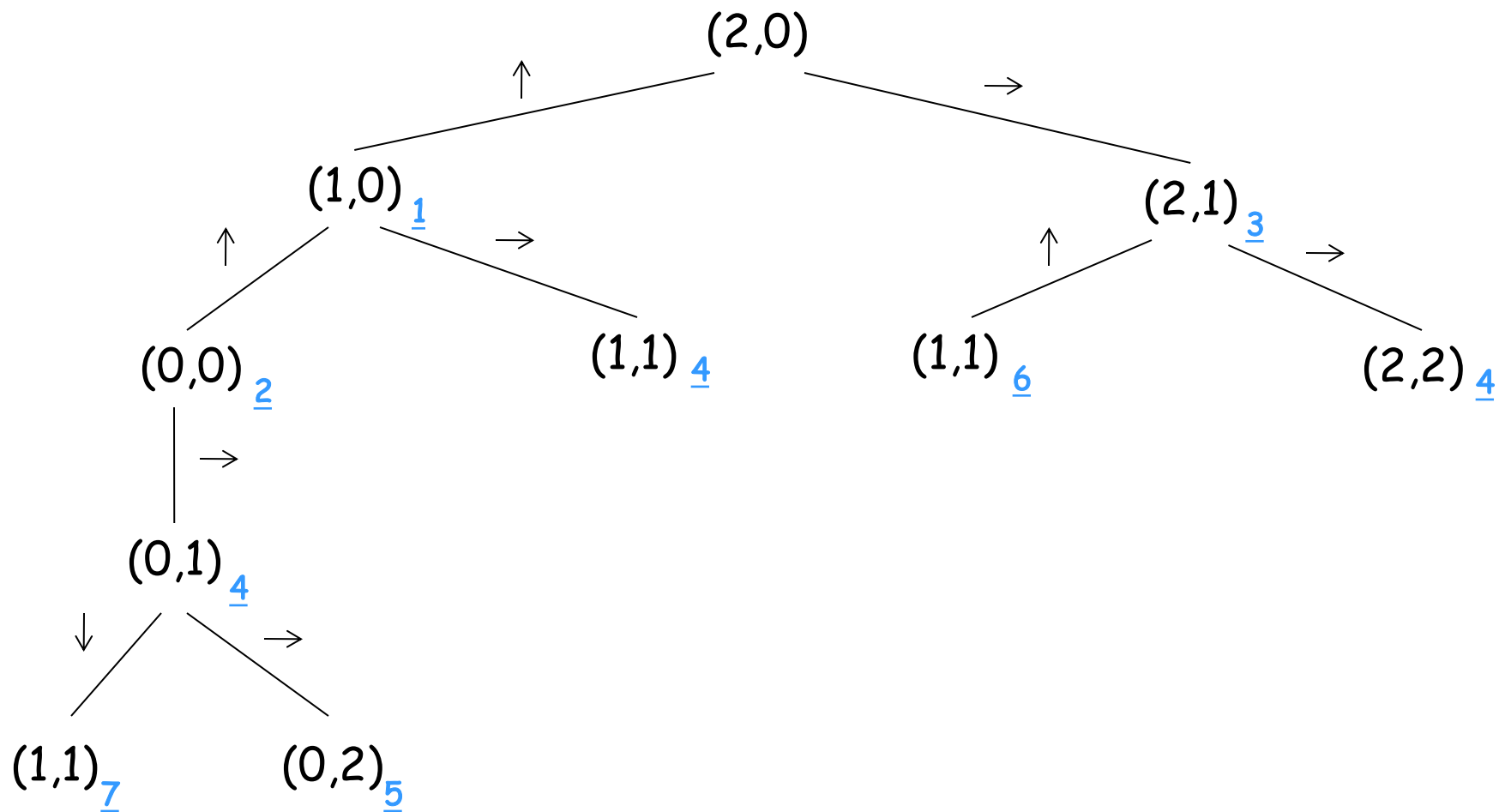
	0	1	2	3
0				
1				Fin
2	Inicio			



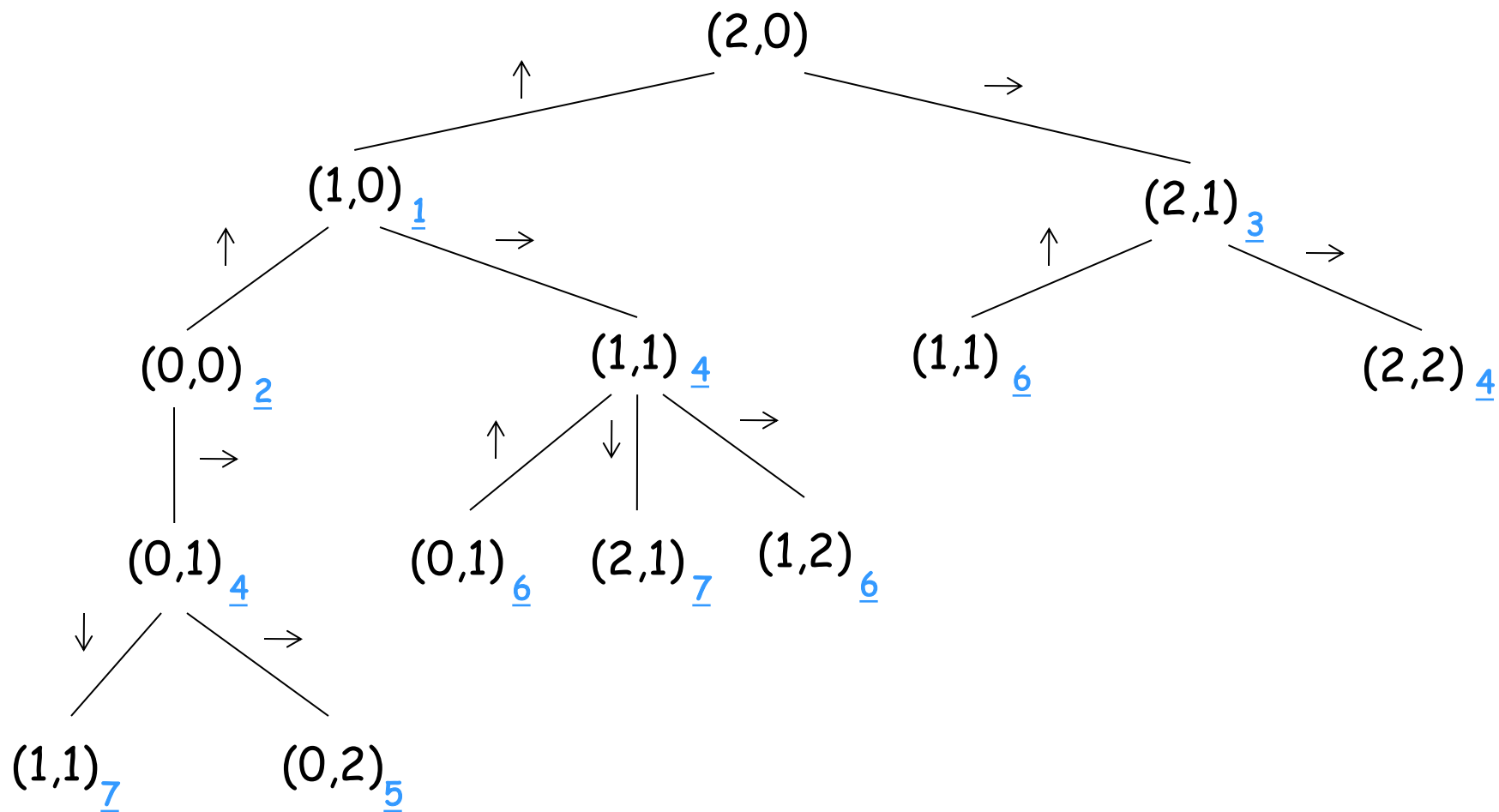
	0	1	2	3
0				
1				Fin
2	Inicio			



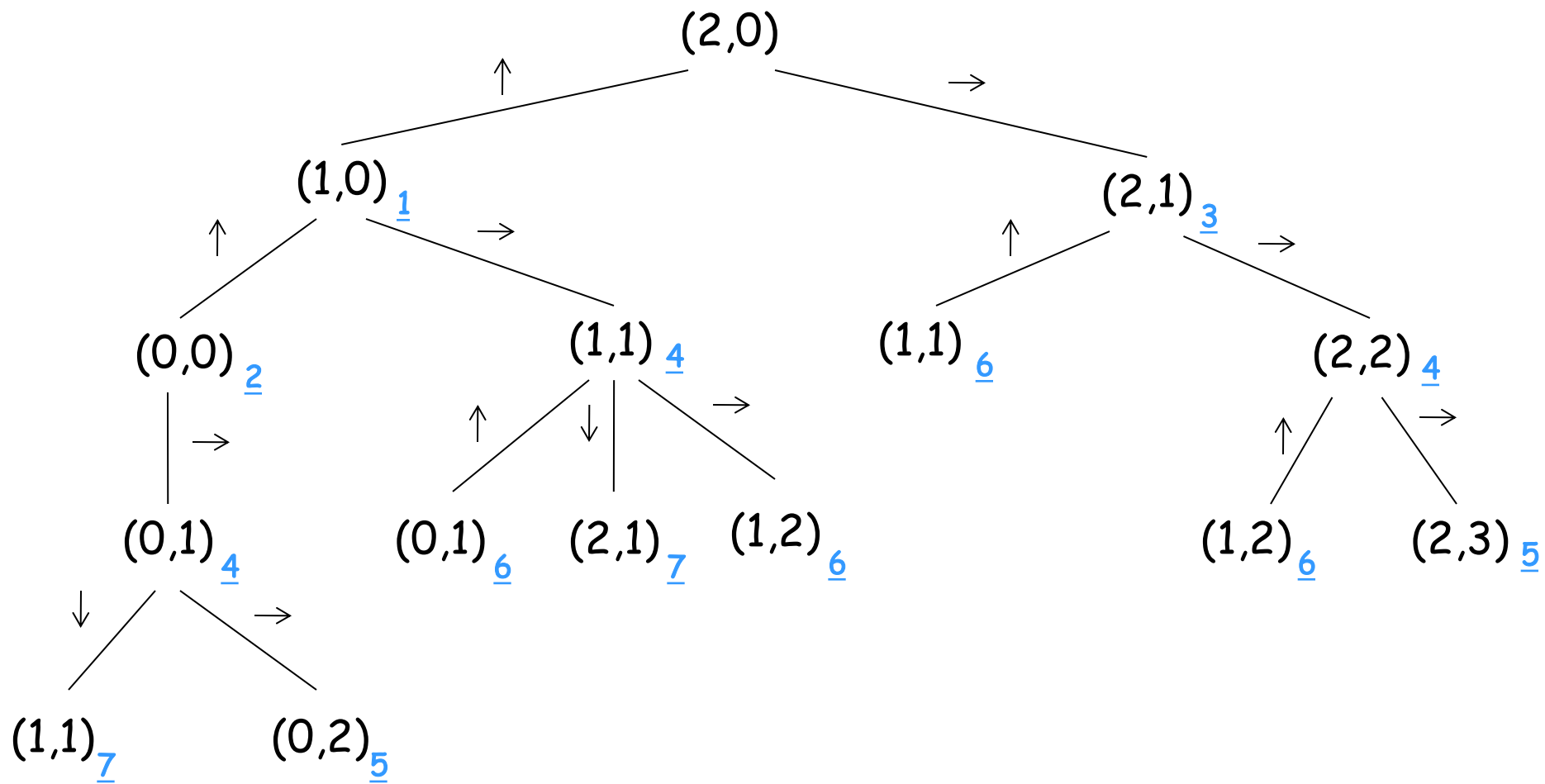
	0	1	2	3
0				
1				Fin
2	Inicio			



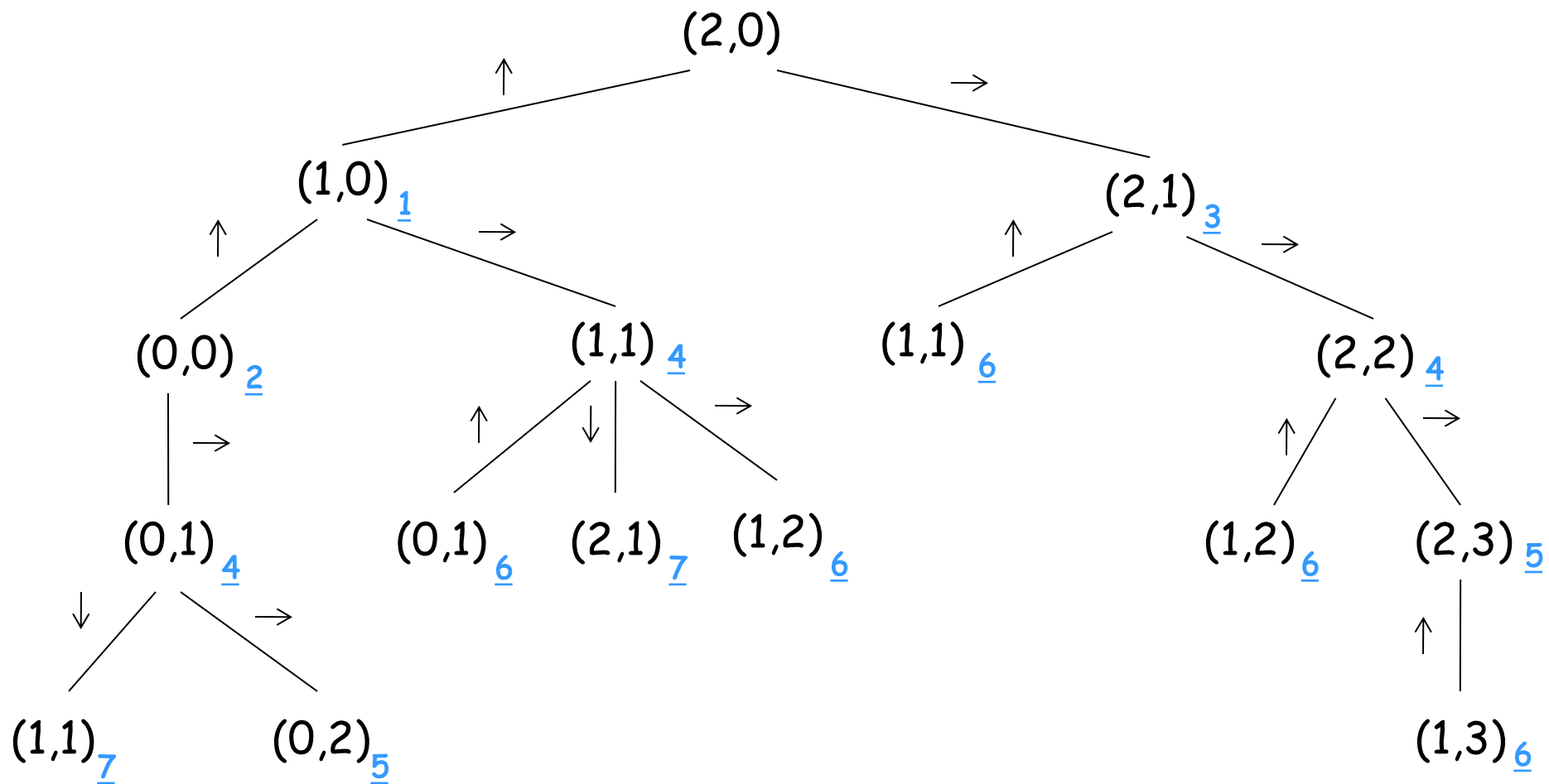
	0	1	2	3
0				
1				Fin
2	Inicio			



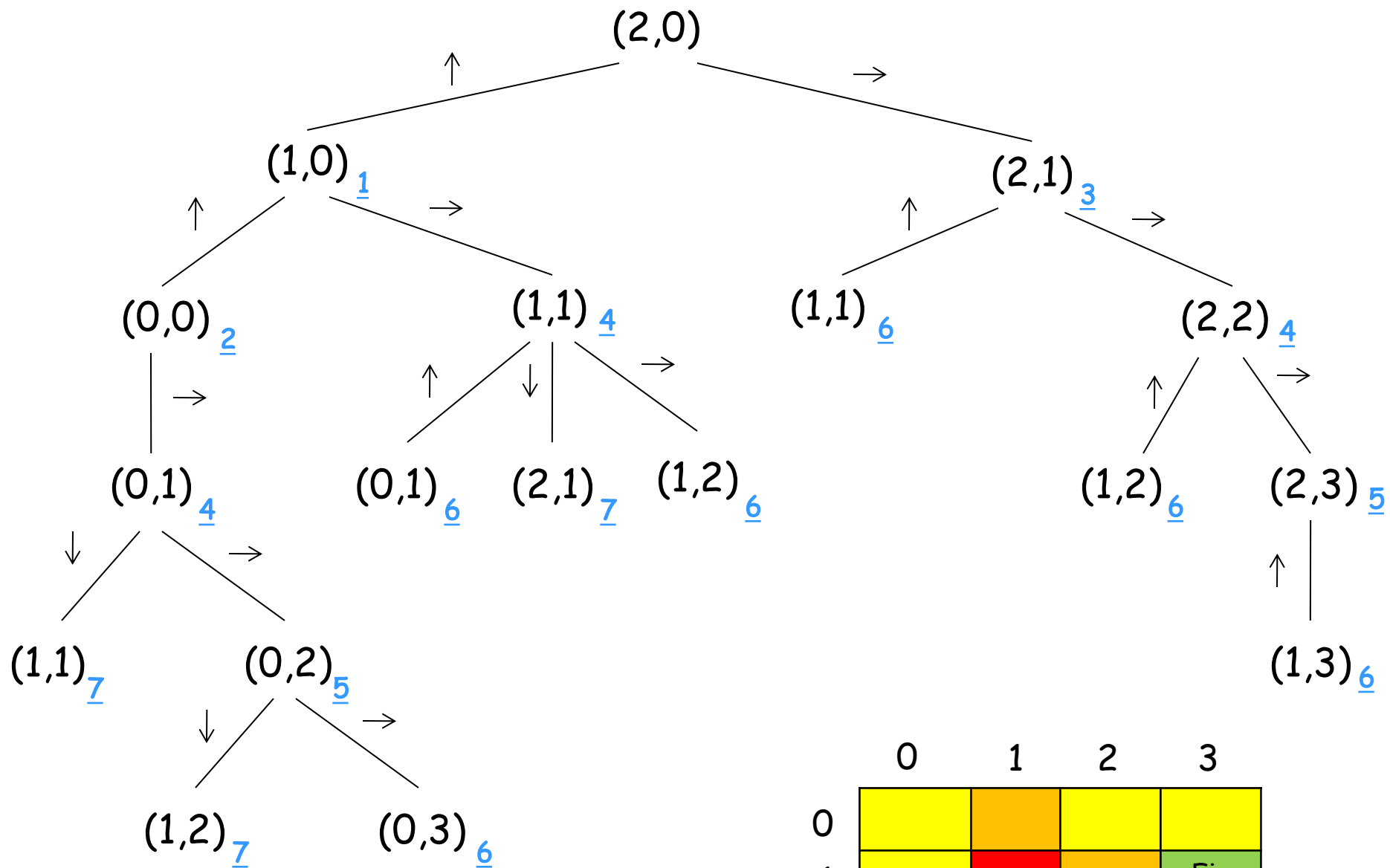
	0	1	2	3
0				
1				Fin
2	Inicio			



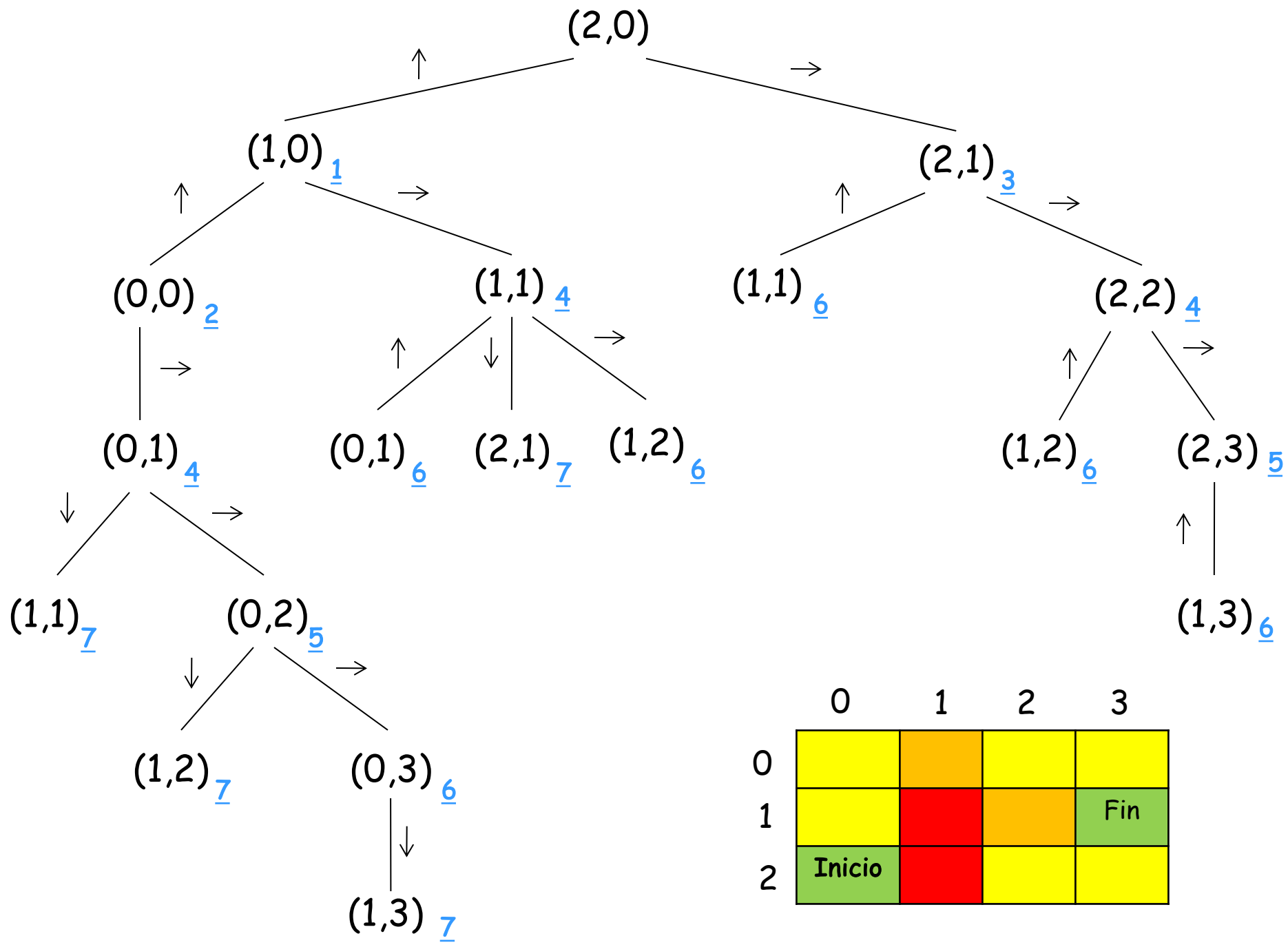
	0	1	2	3
0				
1				Fin
2	Inicio			

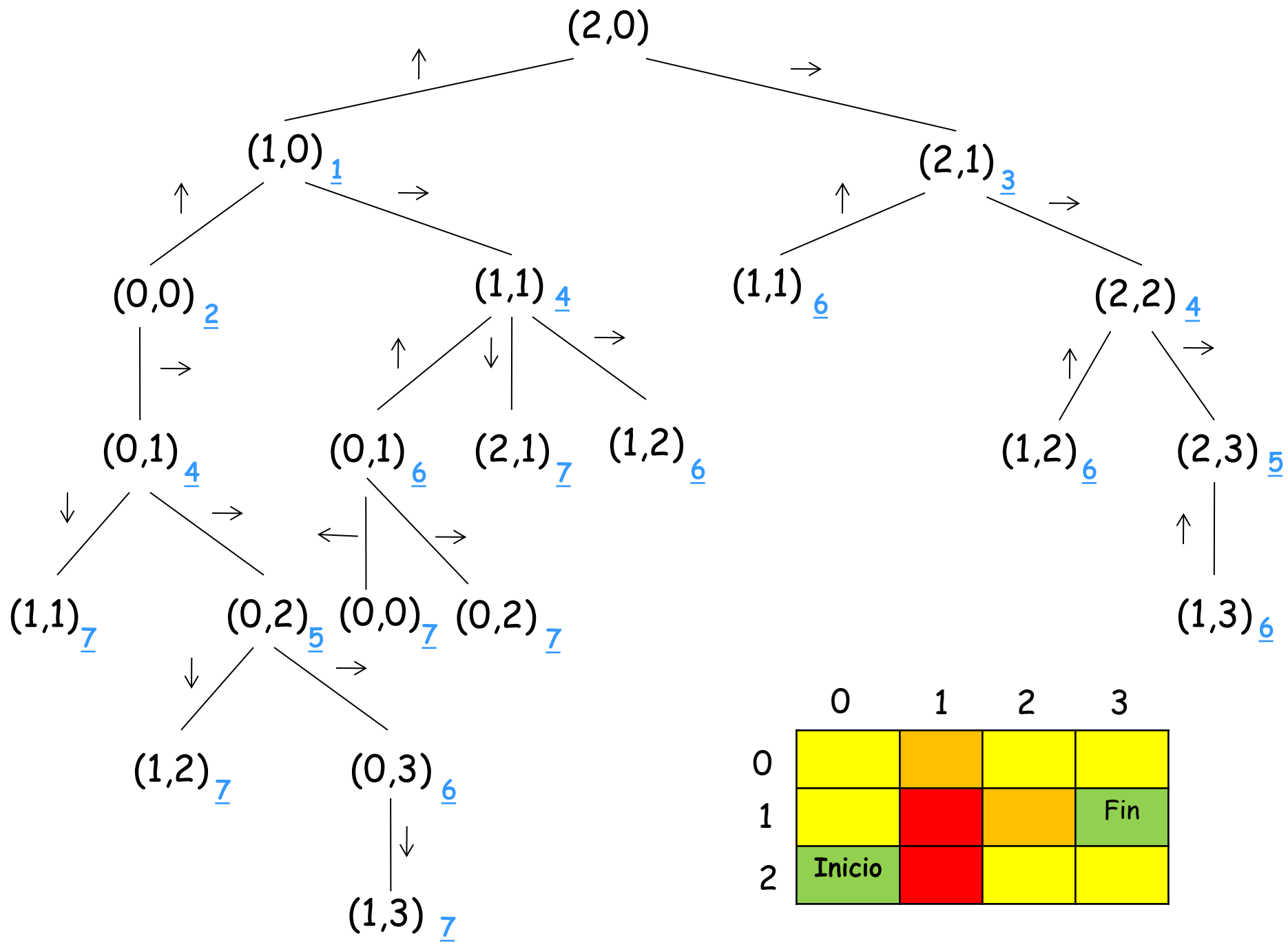


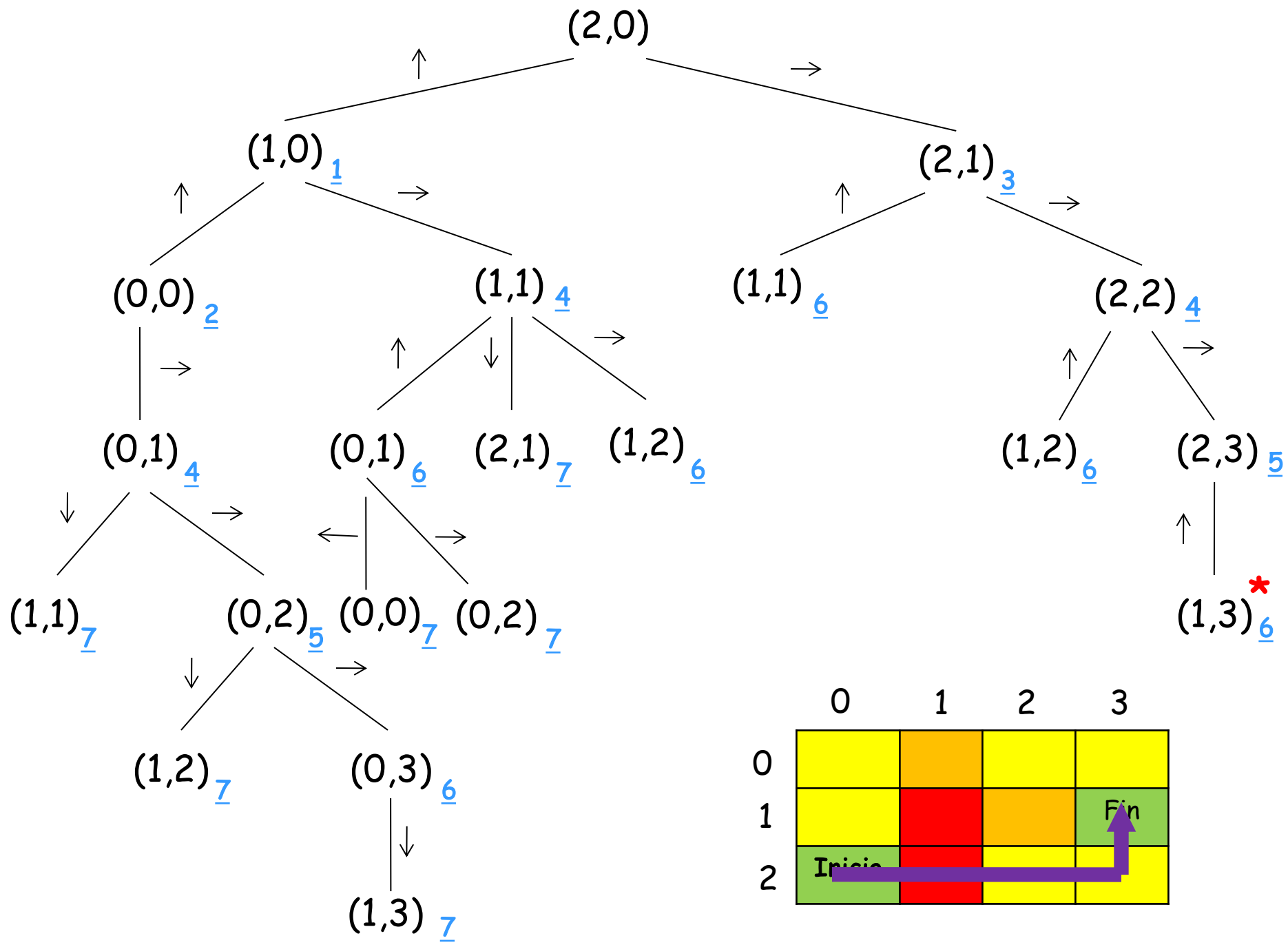
	0	1	2	3
0				
1				Fin
2	Inicio			



	0	1	2	3
0				
1				Fin
2	Inicio			










	0	1	2	3
0				
1				Fin
2	Inicio			

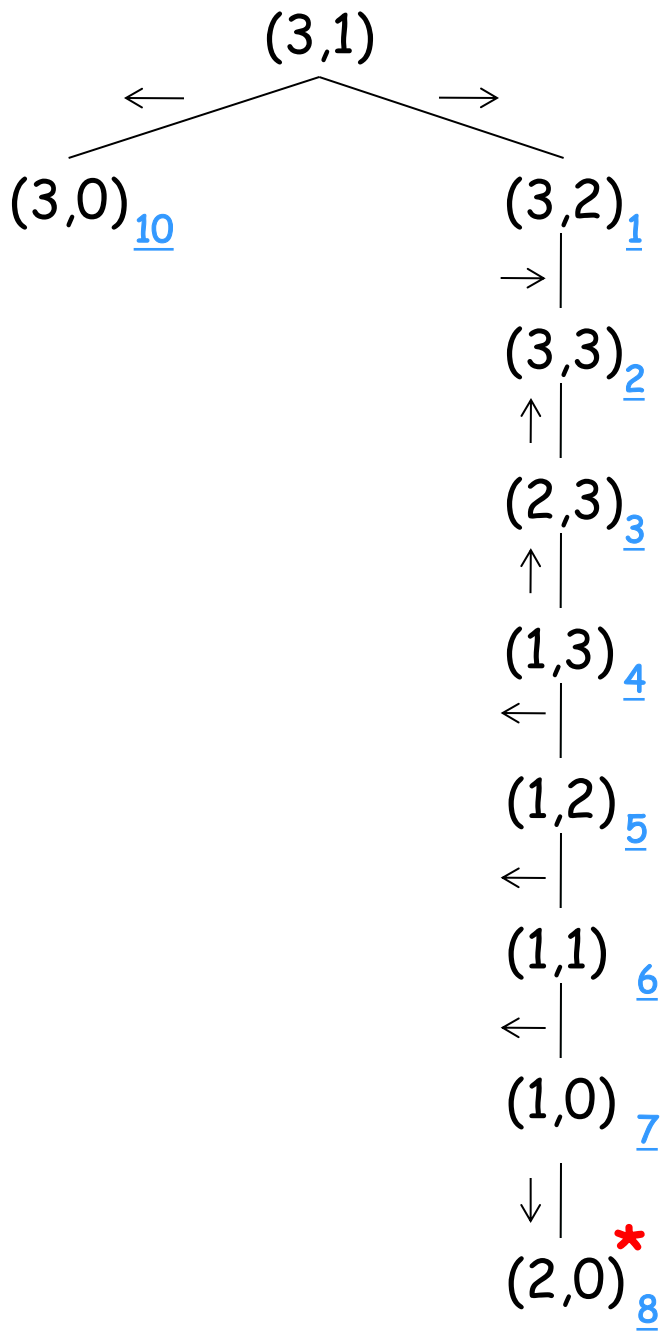
Búsqueda no informada




Aplicar búsqueda por costo uniforme

- El costo de desplazarse hacia una casilla donde no hay gato es 1
- El costo de desplazarse hacia una casilla con gato es 10

* Evite devolverse

	0	1	2	3
0				
1				
2				
3				







	0	1	2	3
0				
1				
2				
3				

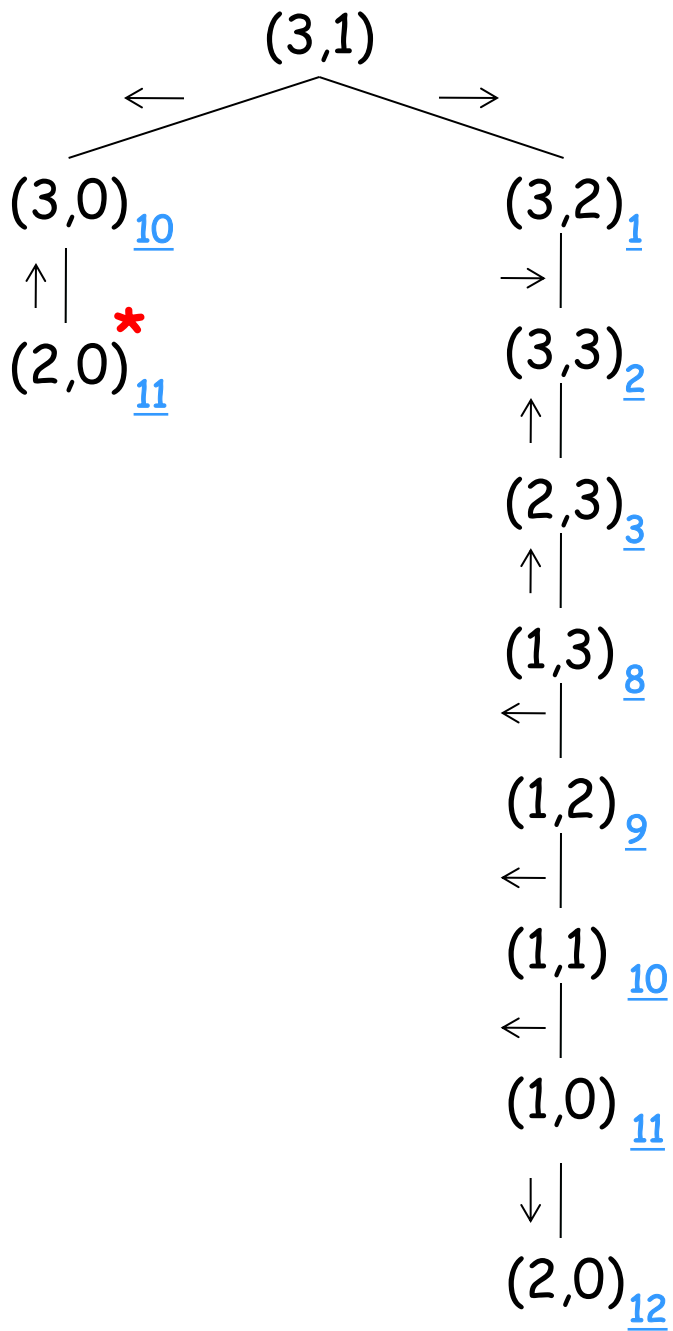
Búsqueda no informada





Aplicar búsqueda por costo uniforme

- El costo de desplazarse hacia una casilla donde no hay gato ni ratonera es 1
- El costo de desplazarse hacia una casilla con gato es 10
- El costo de desplazarse hacia una casilla con ratonera es 5

* Evite devolverse

	0	1	2	3
0				
1				
2				
3				



	0	1	2	3
0				
1				
2				
3				

Búsqueda no informada

Aplicar búsqueda de
costo uniforme

Agente Minero. Indique el
camino que sigue el agente. El
costo de cada movimiento es 1.

 Punto de inicio

 Lingote

	0	1	2
0			
1			
2			
3			

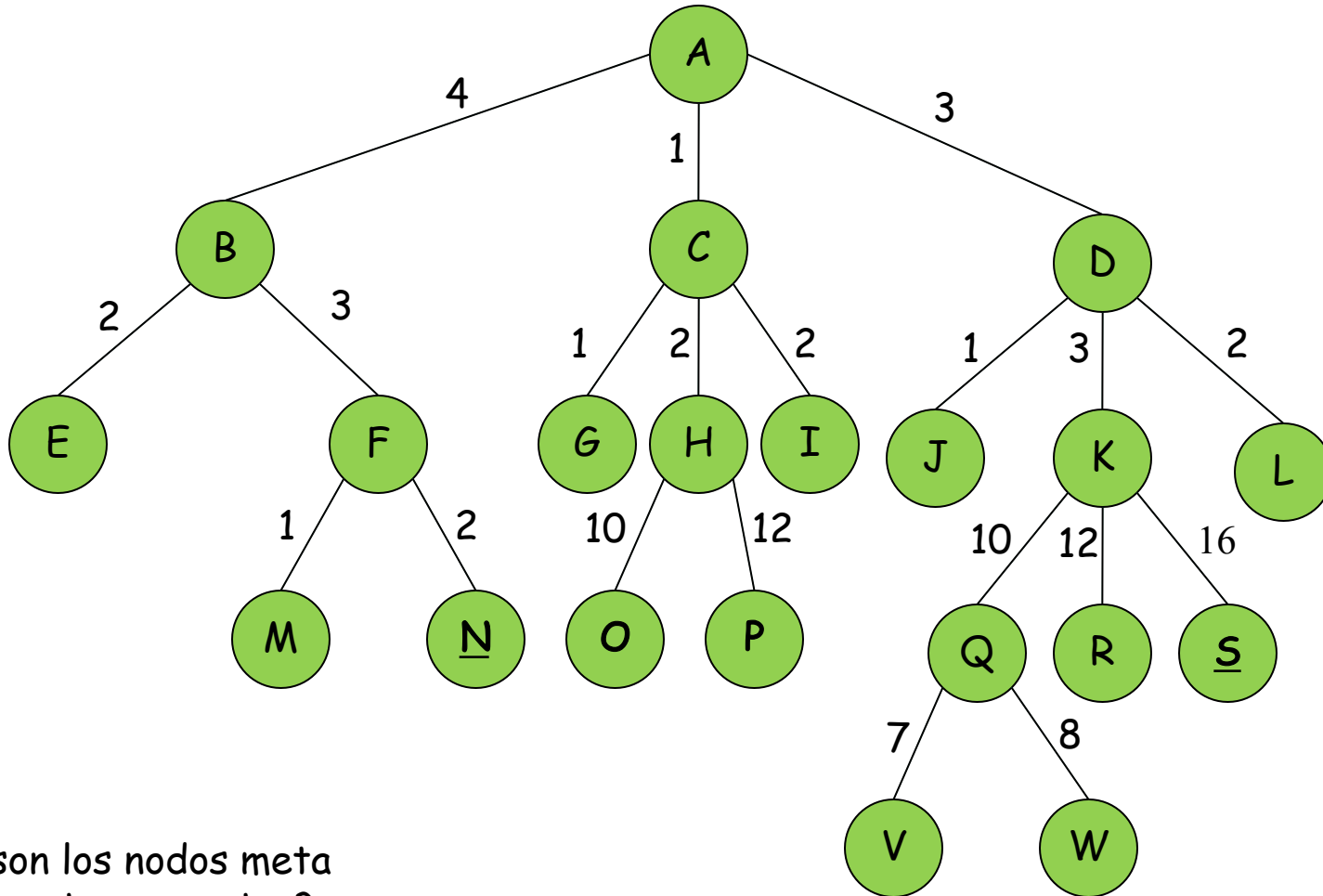
(3,2)

↑
(2,2) 1

* No evite devolverse

Búsqueda no informada

Aplicar búsqueda de costo uniforme

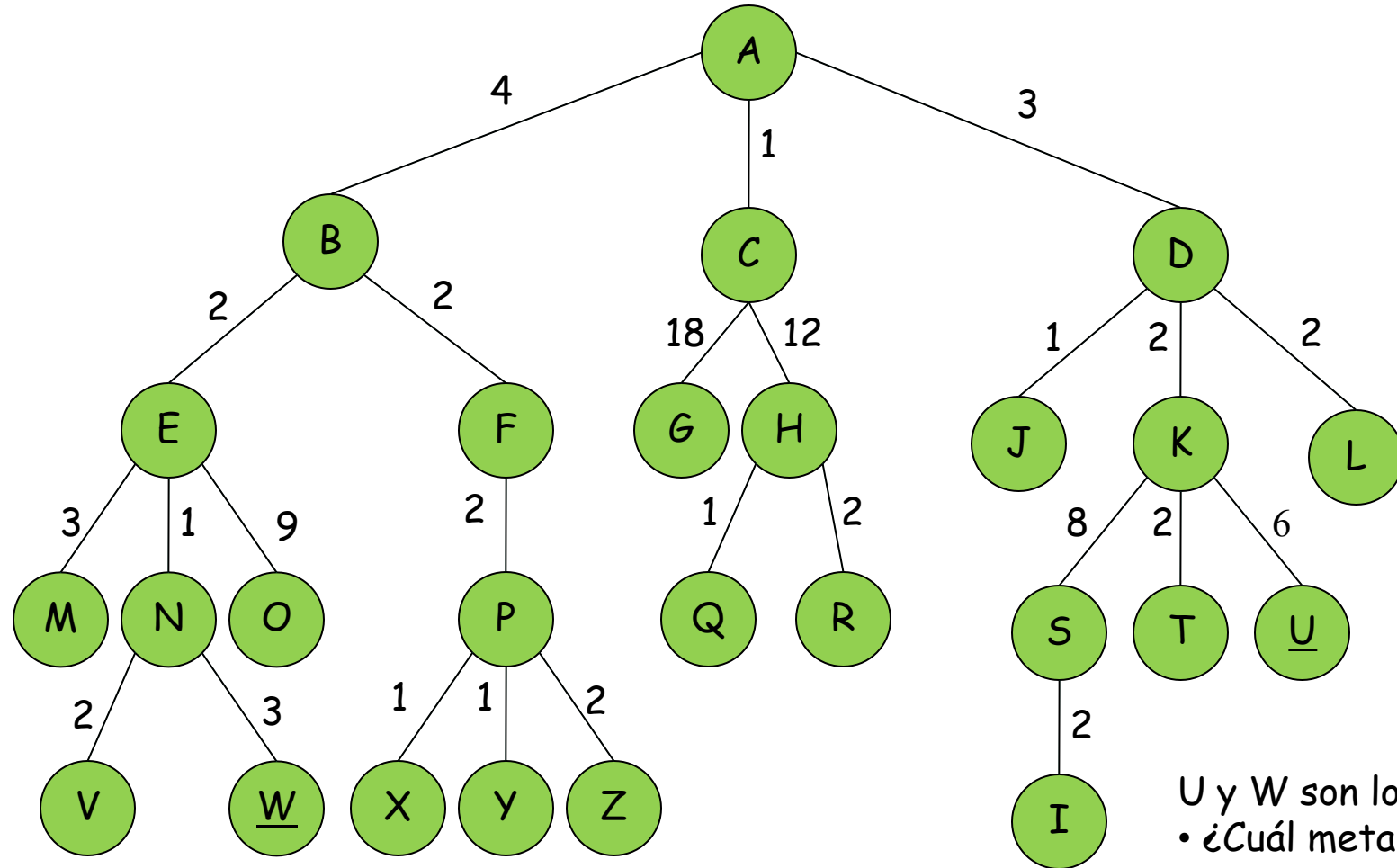


N y S son los nodos meta

- ¿Cuál meta encuentra?
- Liste los nodos creados
- Liste los nodos expandidos

Búsqueda no informada

Aplicar búsqueda de costo uniforme



U y W son los nodos meta

- ¿Cuál meta encuentra?
- Liste los nodos creados
- Liste los nodos expandidos





Búsqueda no informada













Aplicar búsqueda de costo uniforme

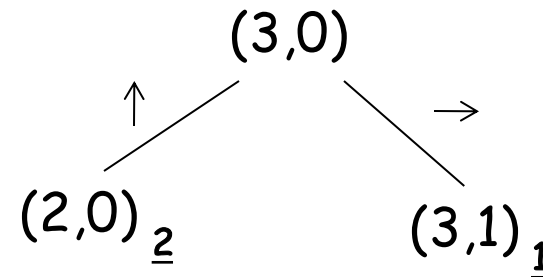
Operadores

- Arriba
- Abajo
- Izquierda
- Derecha

Los costos varían de acuerdo al color de la casilla a la cual se llegue

-  : 1
-  : 2
-  : 3
-  : 1

	0	1	2
0			 Fin
1			
2			
3	 Inicio		



* Evite devolverse




Búsqueda no informada







Aplicar búsqueda de costo uniforme

Operadores

- Arriba
- Abajo
- Izquierda
- Derecha

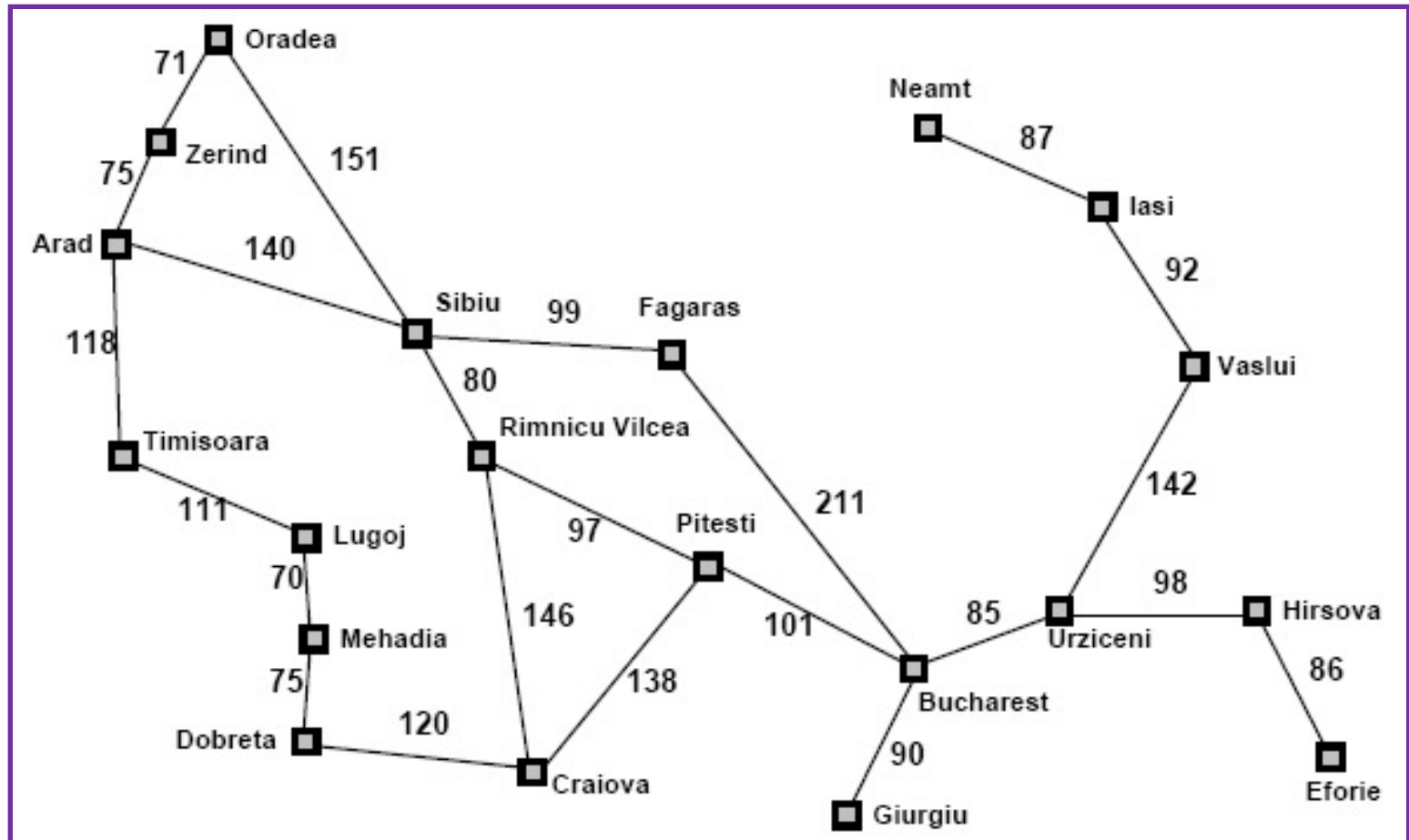
Los costos varían de acuerdo al color de la casilla a la cual se llegue

-  : 1
-  : 2
-  : 3

				Fin
				
				
				
Inicio				

Búsqueda no informada

Aplicar búsqueda de costo uniforme
para ir de Arad a Bucarest



Búsqueda no informada

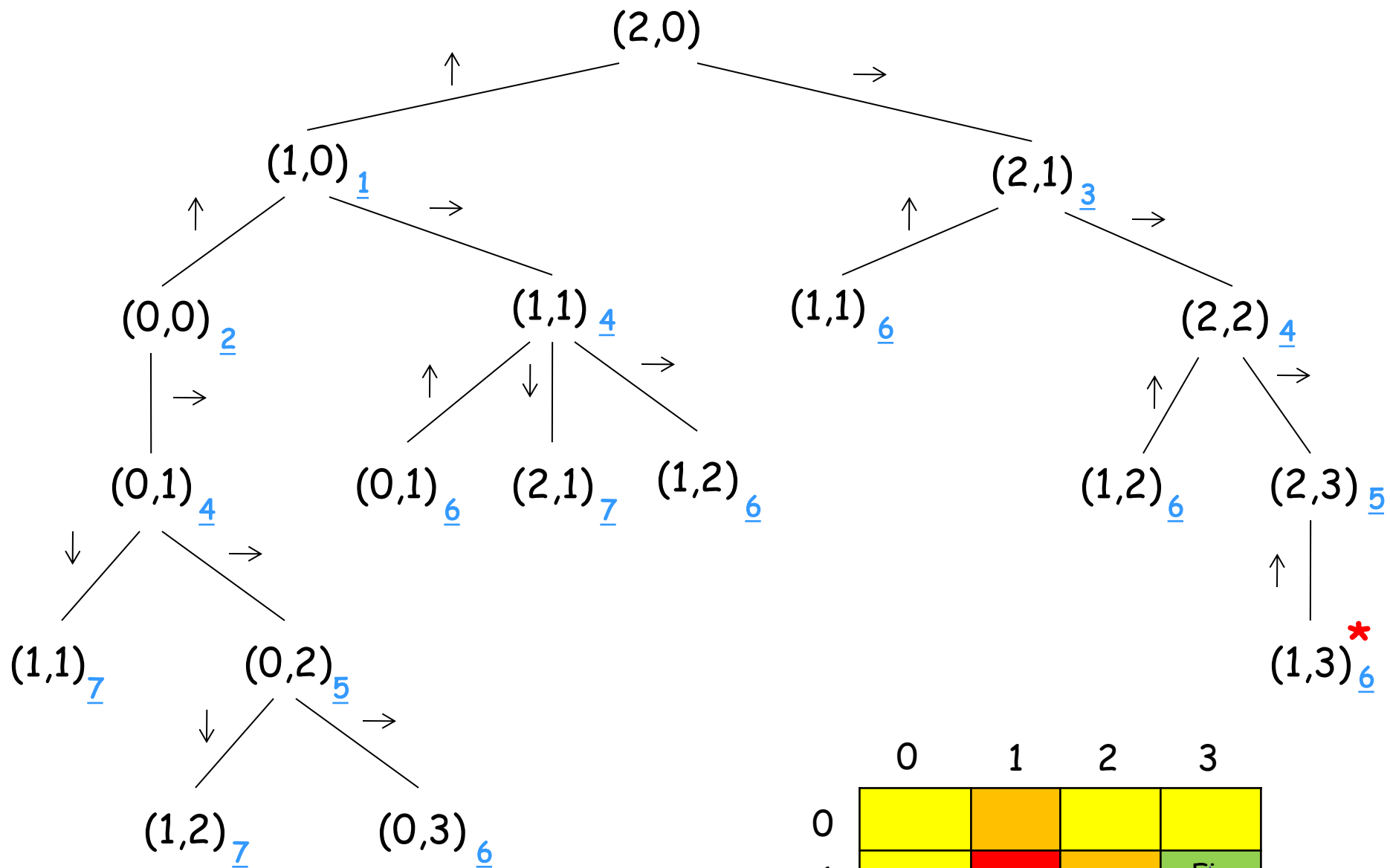
Búsqueda de costo uniforme

- **Complejidad:** si
- **Complejidad temporal:** número de nodos con costo menor al de la solución óptima
- **Complejidad espacial:** número de nodos con costo menor al de la solución óptima
- **Solución óptima:** si

Búsqueda no informada

Implementación

- Se puede implementar considerando la lista de nodos a expandir como una cola de prioridad, donde la prioridad es el costo y se selecciona aquel con mejor prioridad



	0	1	2	3
0				
1				Fin
2	Inicio			

Búsqueda no informada

Búsqueda no informada:

- Preferente por amplitud
- De costo uniforme
- Preferente por profundidad
- Limitada por profundidad
- Por profundización iterativa

Búsqueda no informada

Búsqueda preferente por profundidad

- Se expande el nodo raíz
- Siempre se expande, de izquierda a derecha, el nodo con mayor profundidad del árbol

Búsqueda no informada

Búsqueda preferente por profundidad

- Se expande el nodo raíz
- Siempre se expande, de izquierda a derecha, el nodo con mayor profundidad del árbol

Si se llega a un nodo tal que al expandirlo no tiene hijos, se expanden los nodos de niveles más profundos

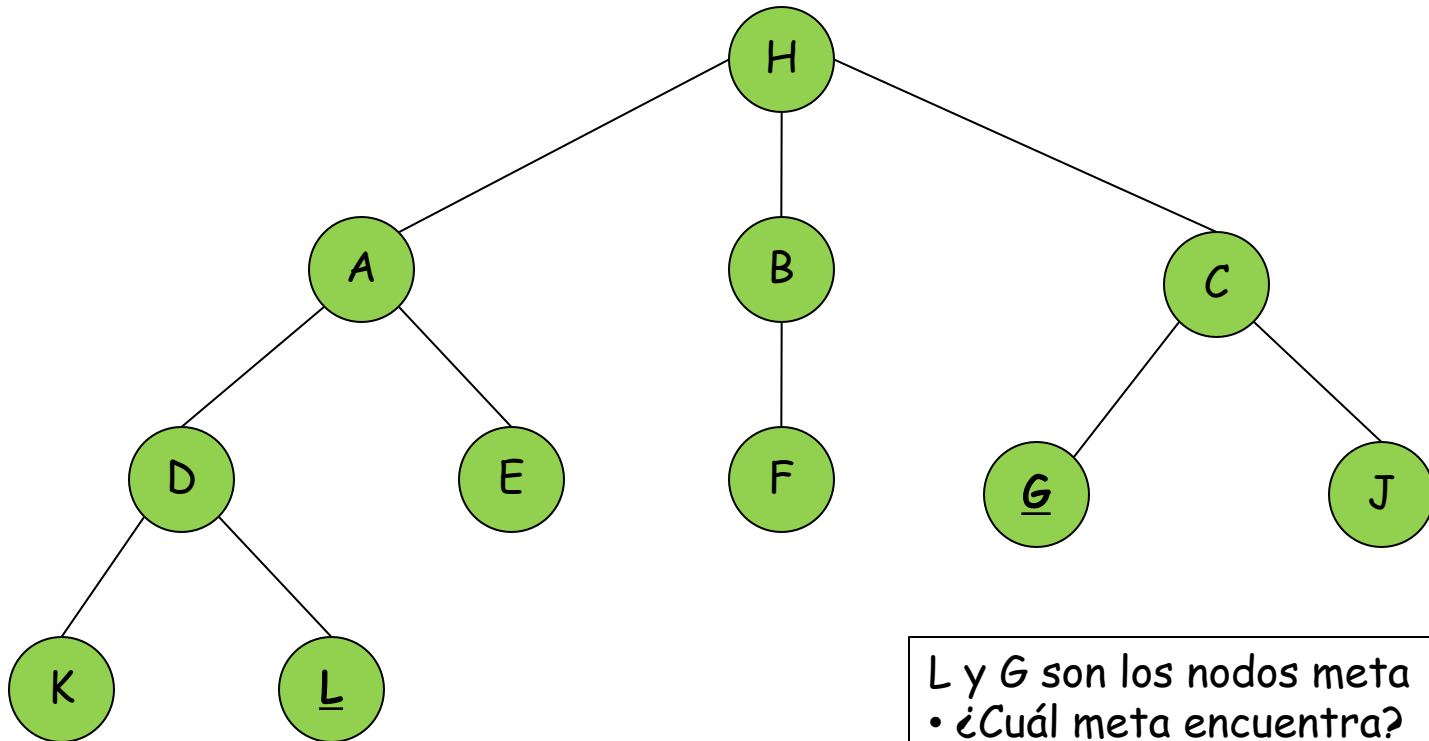
Búsqueda no informada

Búsqueda preferente por profundidad

- En lugar de buscar de forma exhaustiva en cada nivel, este algoritmo sigue las ramas del árbol hacia abajo hasta el nivel más inferior que pueda

Búsqueda no informada

Búsqueda preferente por profundidad

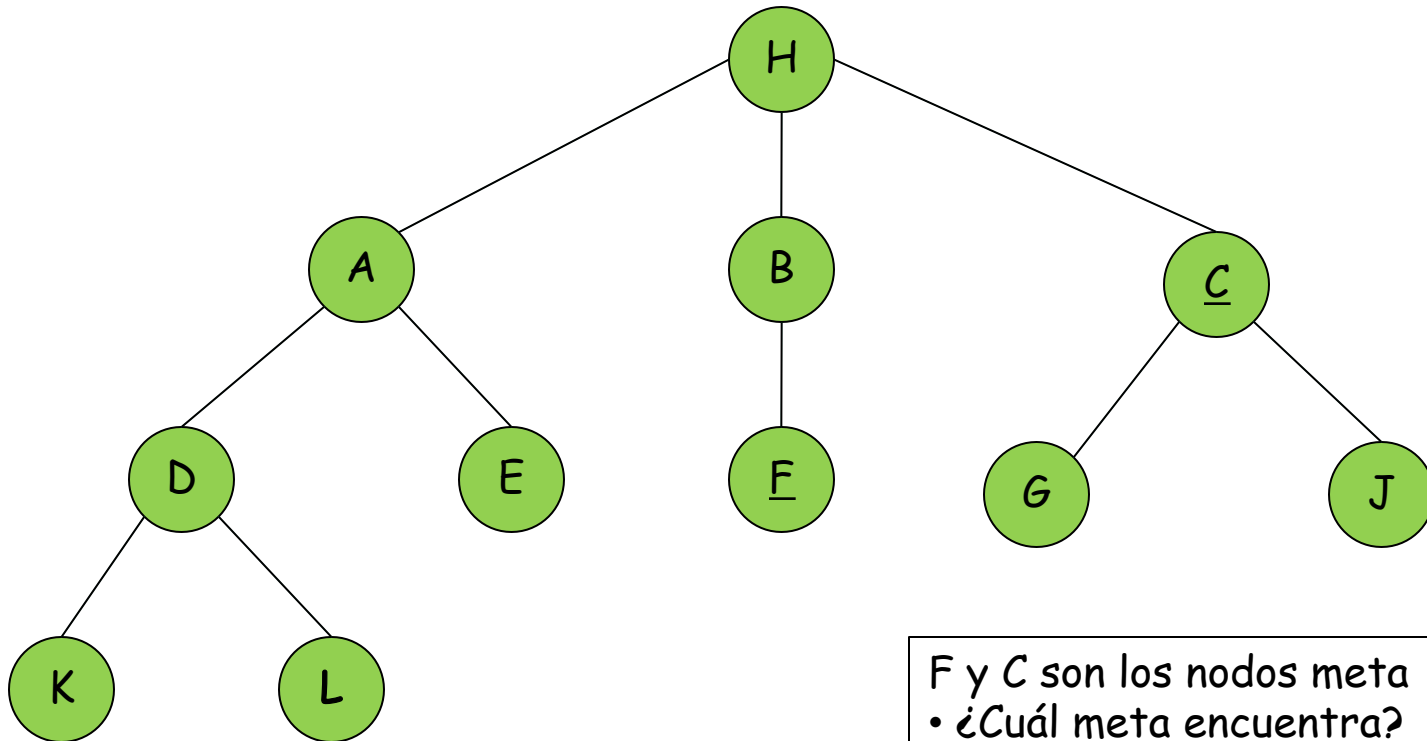


L y G son los nodos meta

- ¿Cuál meta encuentra?
- Liste los nodos creados
- Liste los nodos expandidos

Búsqueda no informada

Búsqueda preferente por profundidad

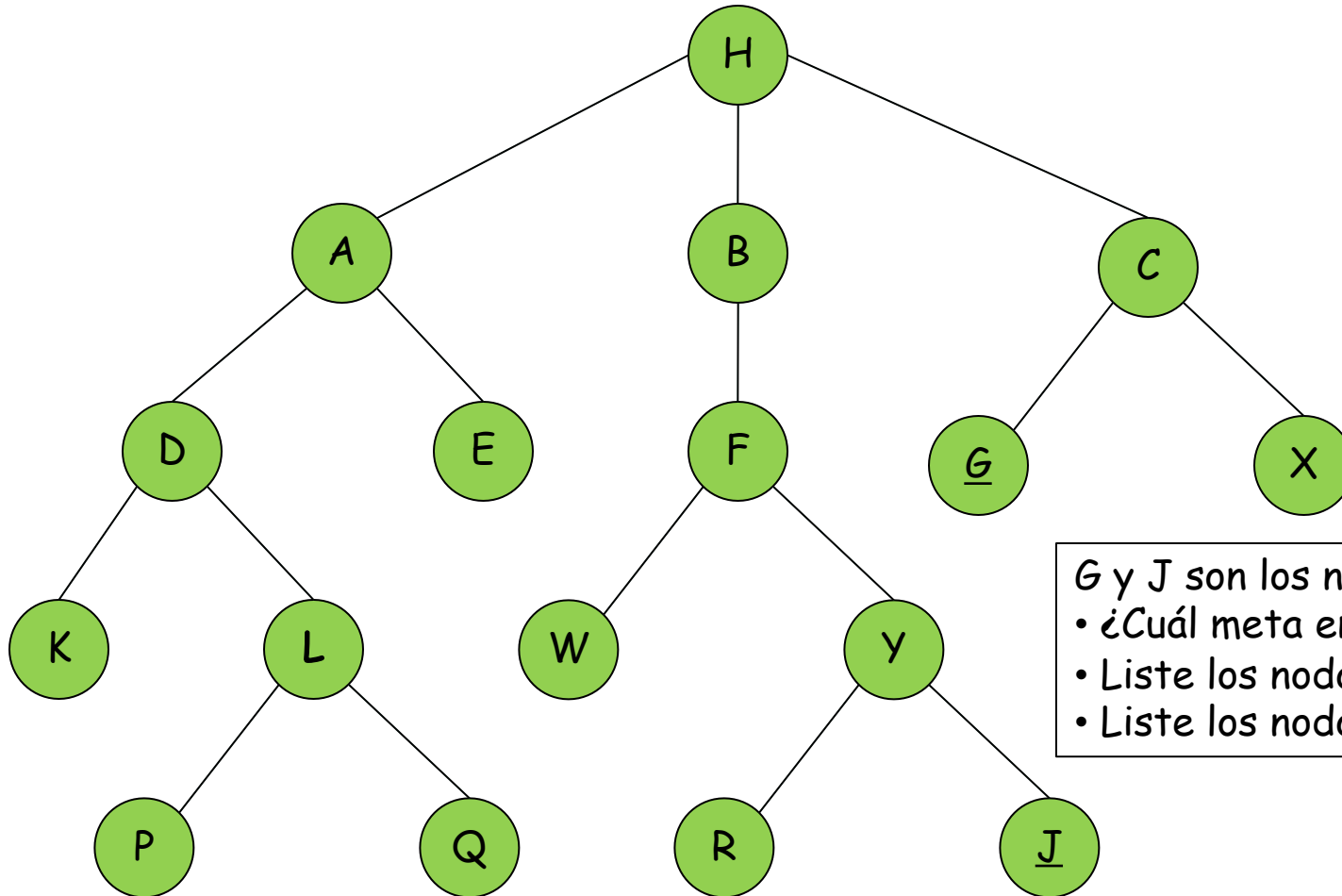


F y C son los nodos meta

- ¿Cuál meta encuentra?
- Liste los nodos creados
- Liste los nodos expandidos

Búsqueda no informada

Búsqueda preferente por profundidad



G y J son los nodos meta

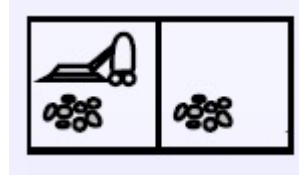
- ¿Cuál meta encuentra?
- Liste los nodos creados
- Liste los nodos expandidos

Búsqueda no informada

Aplicar búsqueda preferente por profundidad

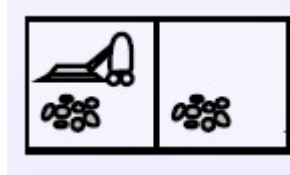
Aplique los operadores en el orden:

- Aspirar
- Izquierda
- Derecha

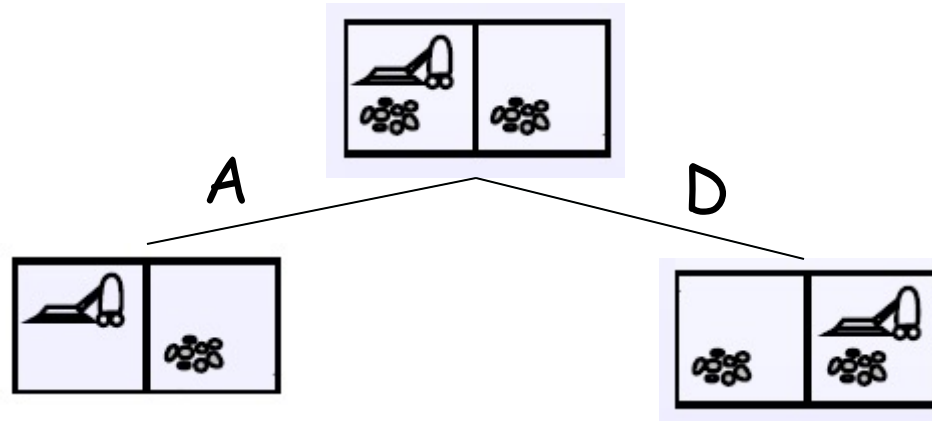


* No evite devolverse

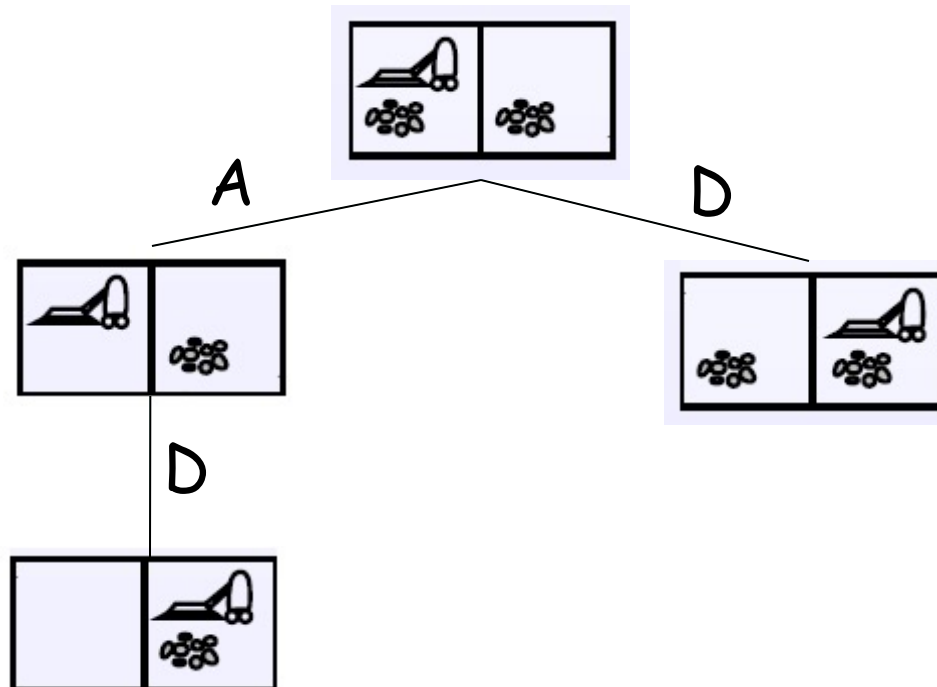
Búsqueda no informada



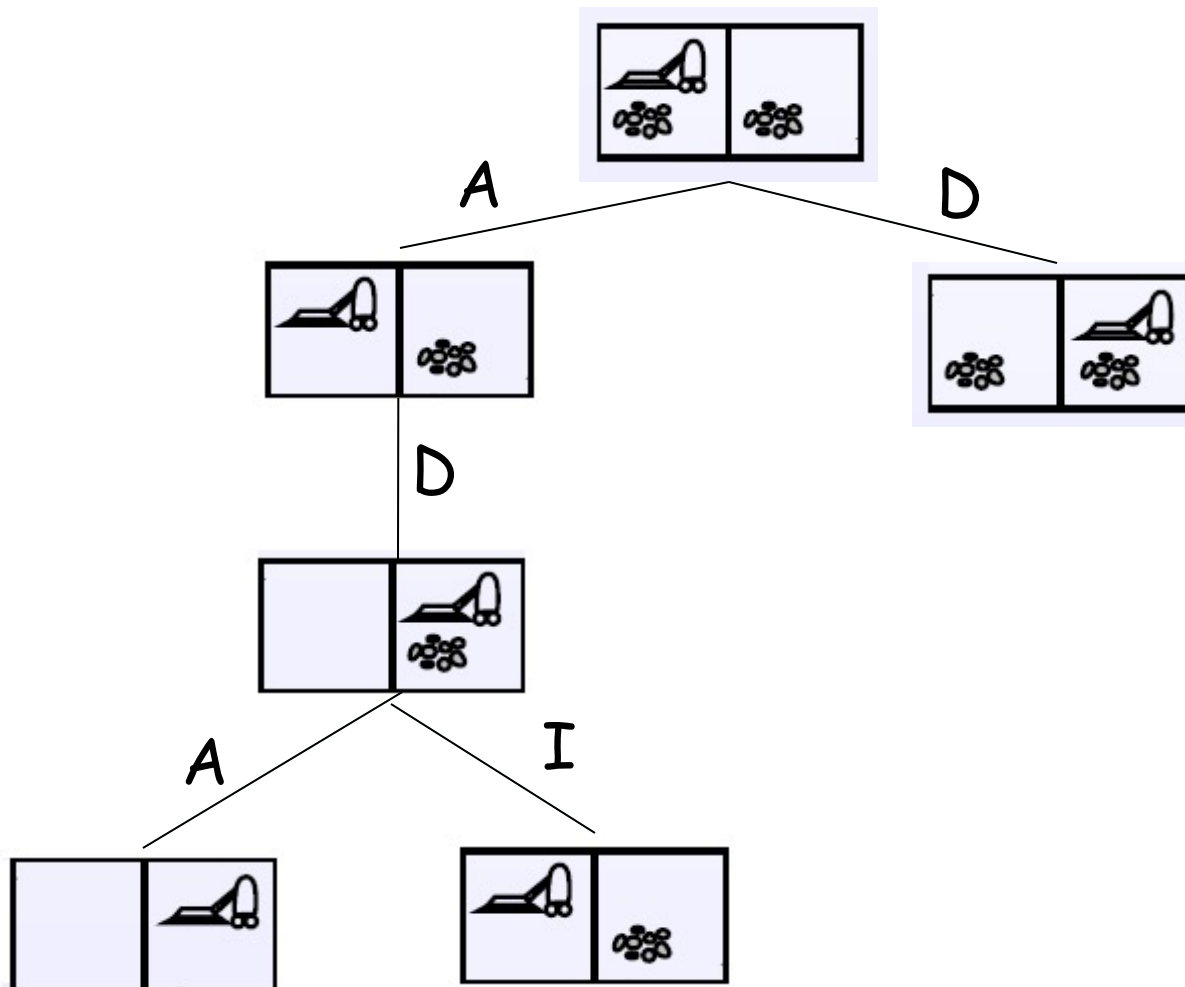
Búsqueda no informada



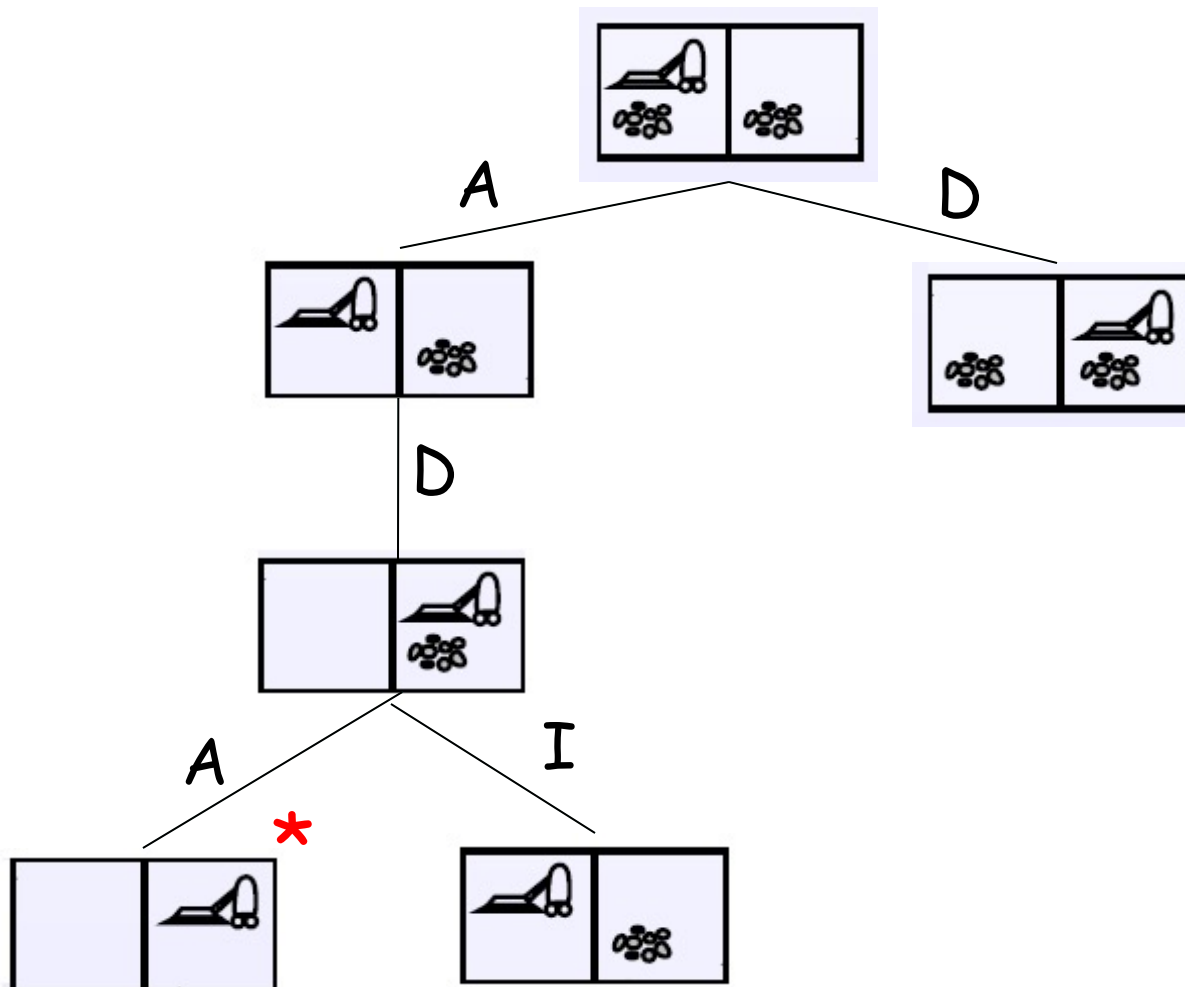
Búsqueda no informada



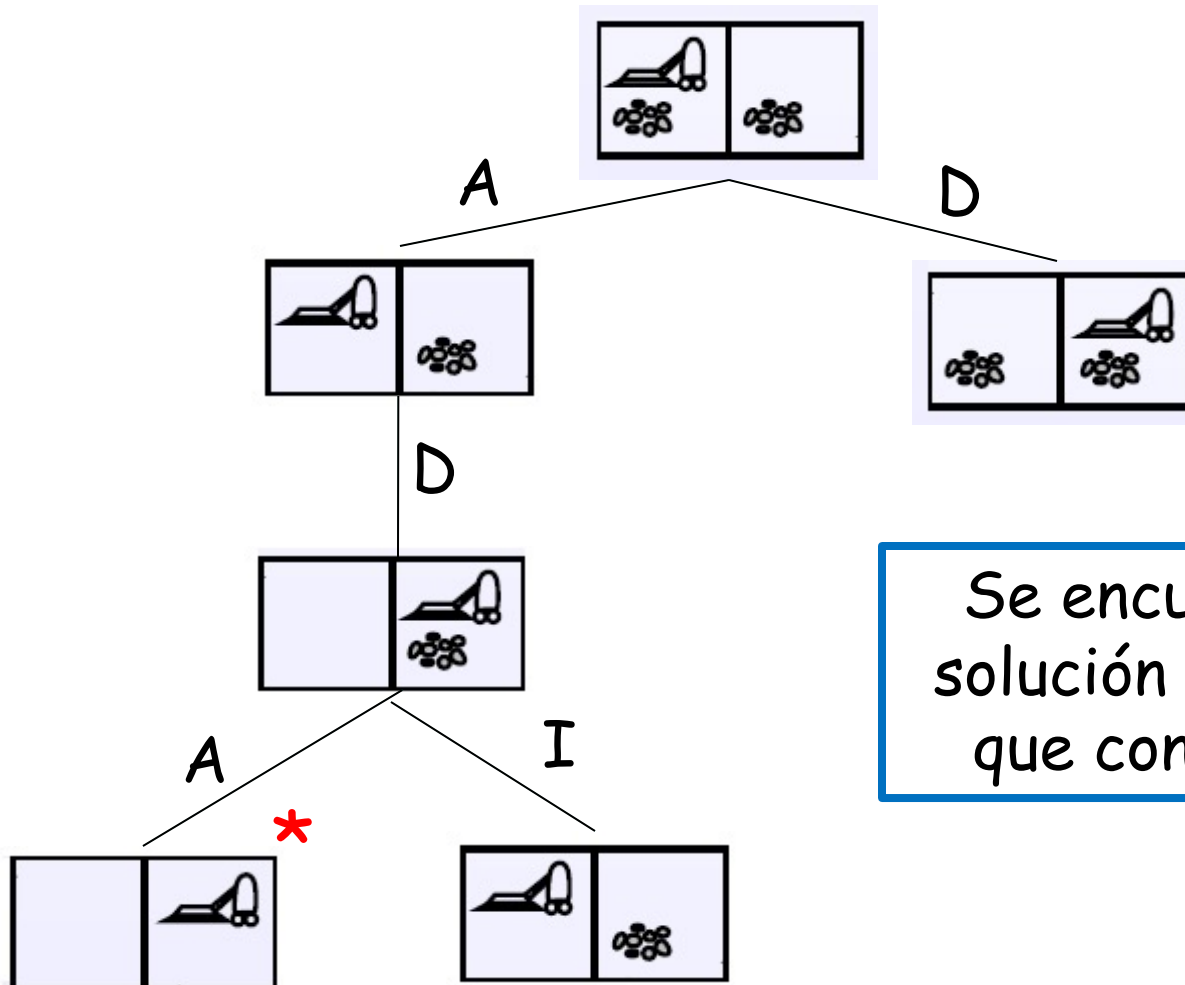
Búsqueda no informada



Búsqueda no informada



Búsqueda no informada



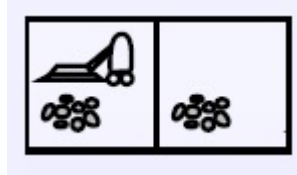
Se encuentra una
solución más rápido
que con amplitud

Búsqueda no informada

Aplicar búsqueda preferente por profundidad

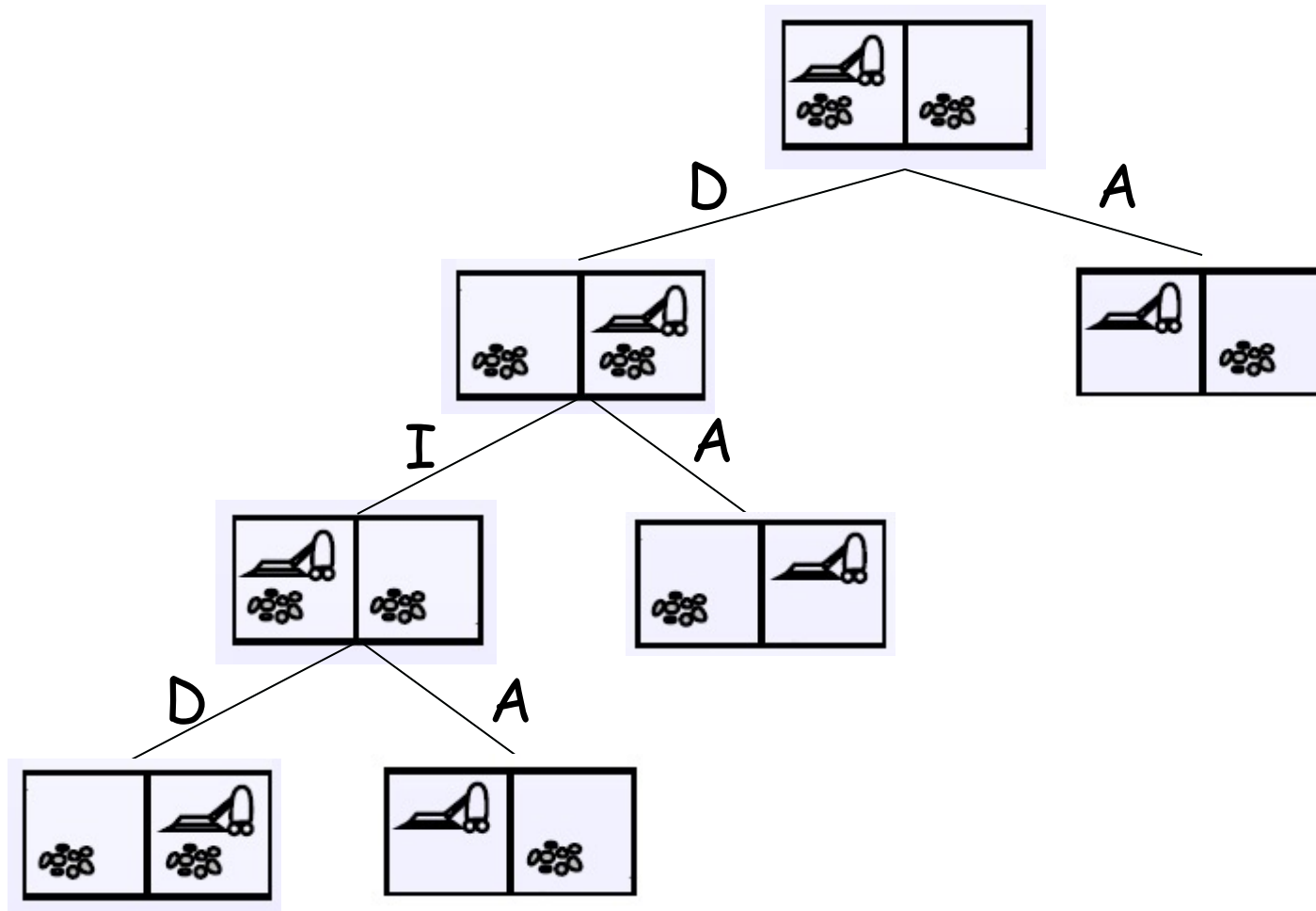
Aplique los operadores en el orden:

- Derecha
- Izquierda
- Aspirar

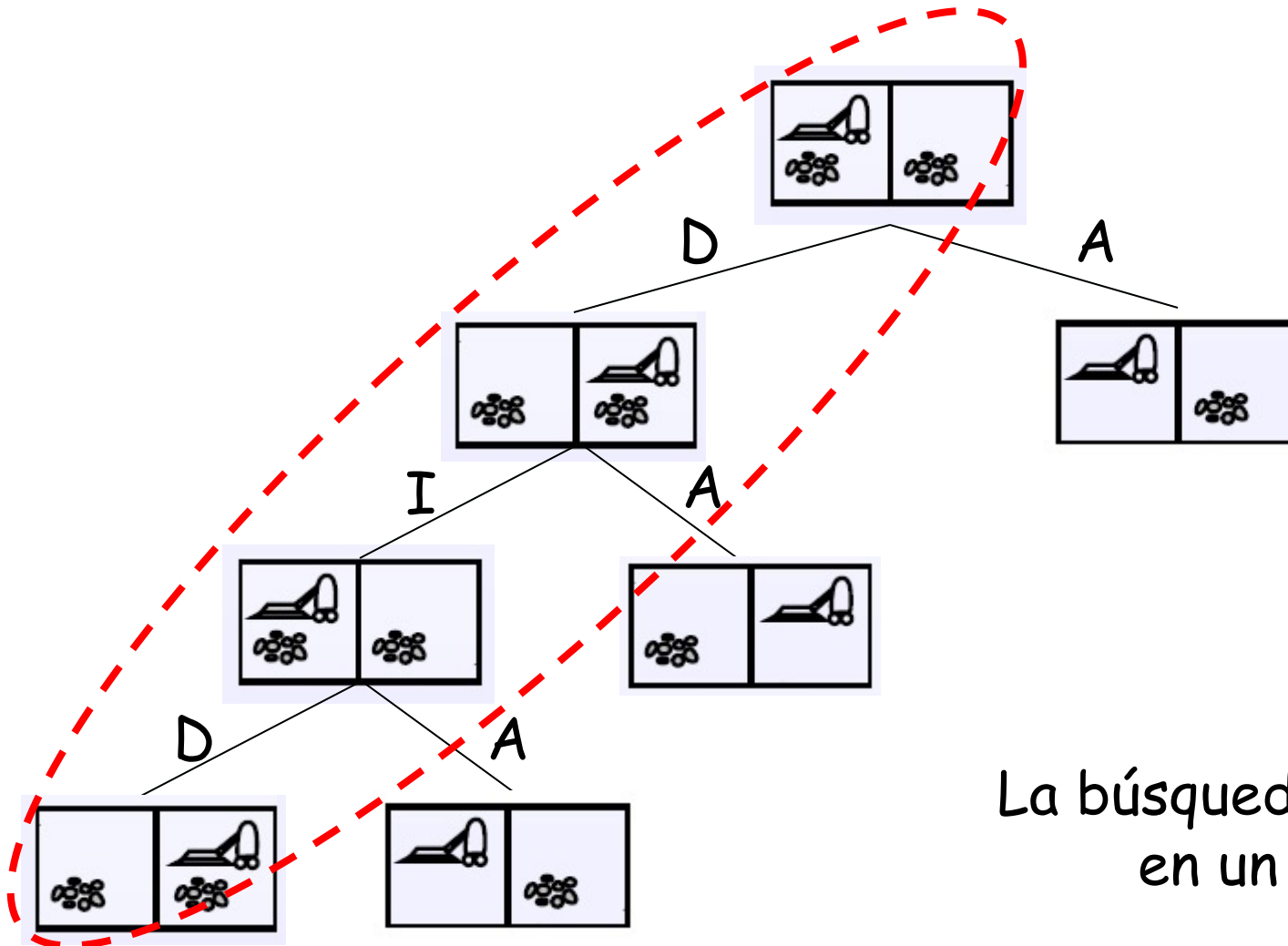


* No evite devolverse

Búsqueda no informada



Búsqueda no informada



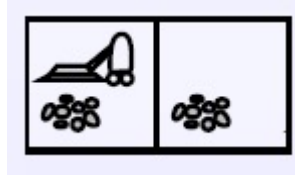
La búsqueda se queda en un ciclo

Búsqueda no informada

Aplicar búsqueda preferente por profundidad

Aplique los operadores en el orden:

- Derecha
- Izquierda
- Aspirar



Profundidad es sensible al orden de los operadores

* No evite devolverse

Búsqueda no informada

Aplicar búsqueda
preferente por
profundidad

Aplique los operadores en
el orden: \rightarrow , \downarrow , \leftarrow , \uparrow

1		2
4	5	3
7	8	6

* No evite devolverse

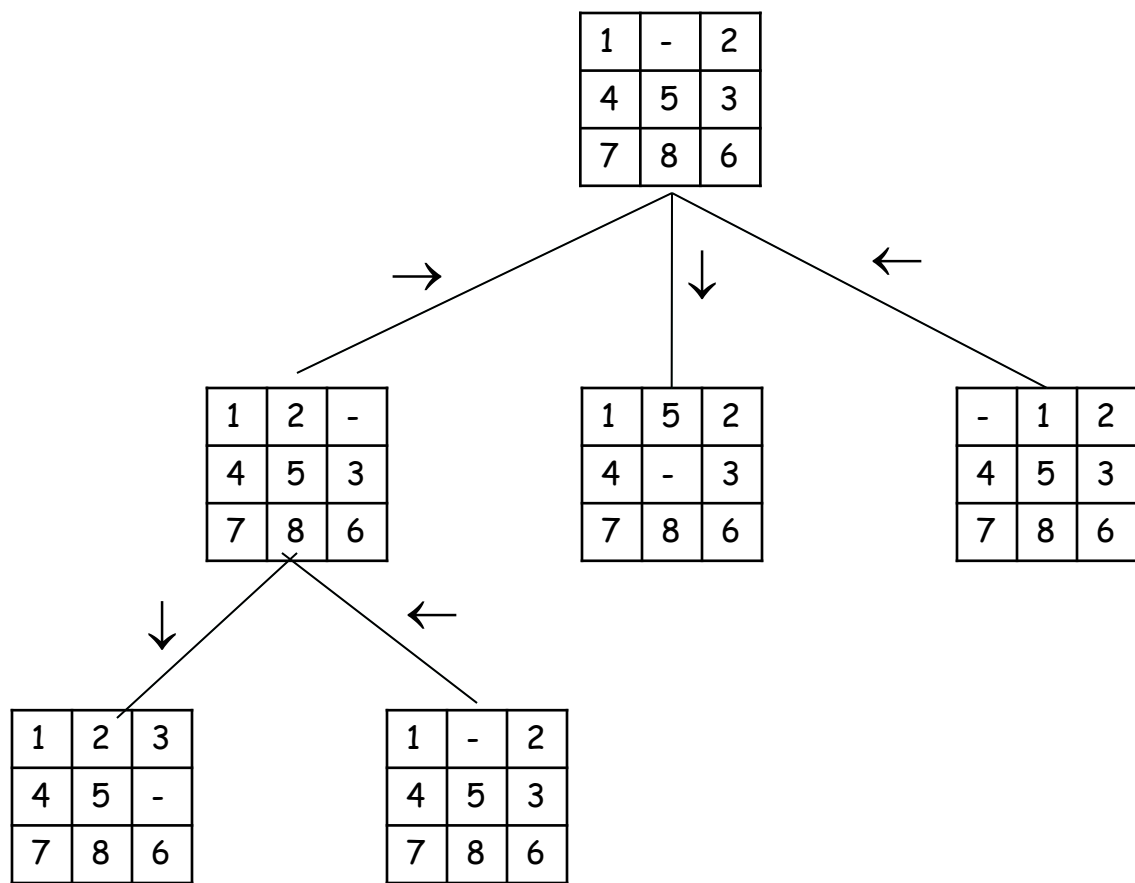
1	-	2
4	5	3
7	8	6



1	2	-
4	5	3
7	8	6

1	5	2
4	-	3
7	8	6

-	1	2
4	5	3
7	8	6



1	-	2
4	5	3
7	8	6



1	2	-
4	5	3
7	8	6

1	5	2
4	-	3
7	8	6

-	1	2
4	5	3
7	8	6



1	2	3
4	5	-
7	8	6

1	-	2
4	5	3
7	8	6



1	2	3
4	5	6
7	8	-

1	2	3
4	-	5
7	8	6

1	2	-
4	5	3
7	8	6

1	-	2
4	5	3
7	8	6



1	2	-
4	5	3
7	8	6

1	5	2
4	-	3
7	8	6

-	1	2
4	5	3
7	8	6



1	2	3
4	5	-
7	8	6

1	-	2
4	5	3
7	8	6



1	2	3
4	5	6
7	8	-

1	2	3
4	-	5
7	8	6

1	2	-
4	5	3
7	8	6

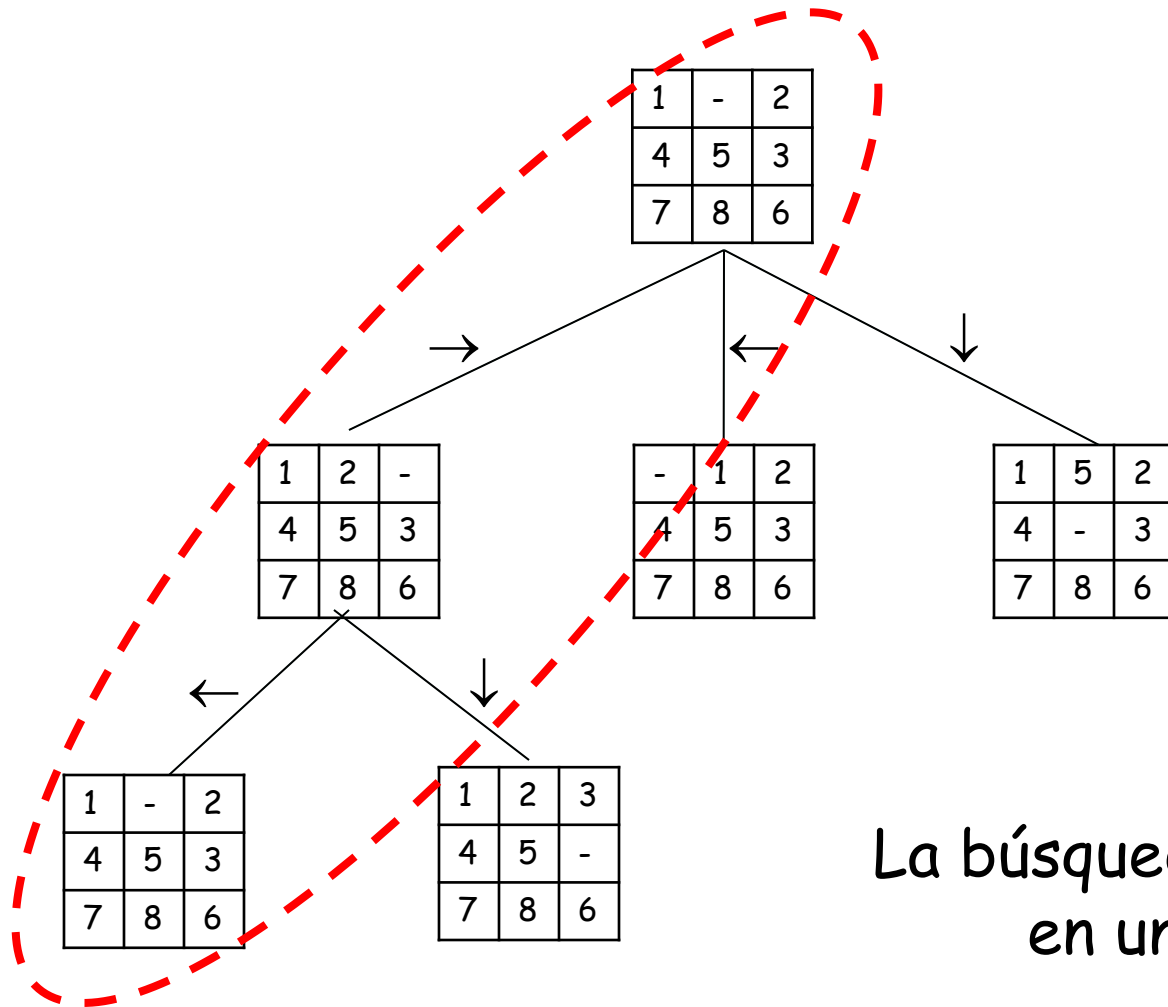
Búsqueda no informada

Aplicar búsqueda
preferente por
profundidad

Aplique los operadores en
el orden: \rightarrow , \leftarrow , \downarrow , \uparrow

1		2
4	5	3
7	8	6

* No evite devolverse






La búsqueda se queda
en un ciclo

Búsqueda no informada

Aplicar búsqueda por profundidad

Aplique los operadores
en el orden $\leftarrow, \uparrow, \rightarrow, \downarrow$

* No evite devolverse




	0	1	2	3
0				
1				
2				
3				

Búsqueda no informada

Aplicar búsqueda por profundidad

Aplique los operadores
en el orden $\rightarrow, \uparrow, \leftarrow, \downarrow$

* No evite devolverse




	0	1	2	3
0				
1				
2				
3				

Búsqueda no informada

Aplicar búsqueda por profundidad

Aplique los operadores
en el orden $\rightarrow, \uparrow, \leftarrow, \downarrow$

* Evite devolverse











	0	1	2	3
0				
1				
2				
3				

Búsqueda no informada

Aplicar búsqueda por profundidad

Aplique los operadores
en el orden $\rightarrow, \uparrow, \leftarrow, \downarrow$

* Evite devolverse

	0	1	2	3
0				
1				
2				
3				

function BUSQUEDA-PREFERENTE-POR-PROFUNDIDAD(problema) returns solución o falla

 Pila pila

 pila = nodo_raiz(problema)

 loop

 if estaVacia(pila) then

 return falló

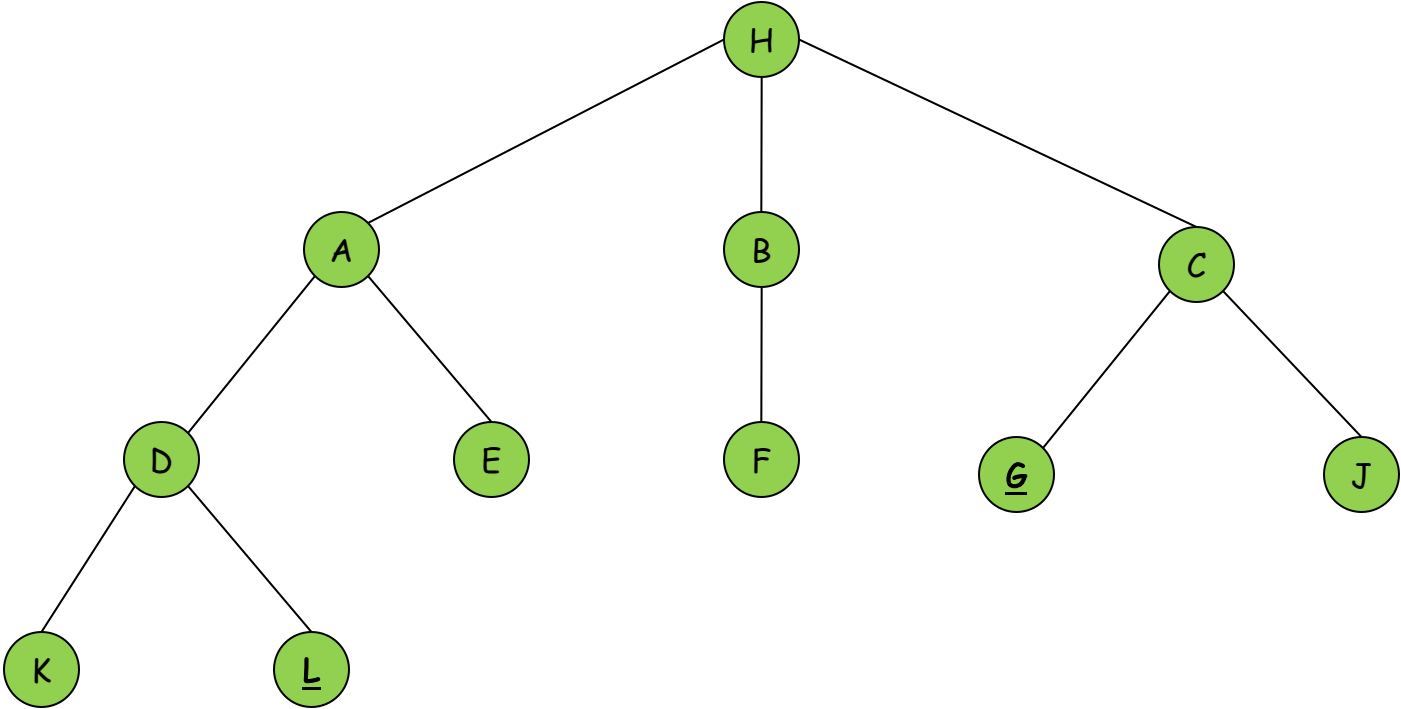
 n=pop(pila)

 if n es un nodo meta then

 return encontró solución

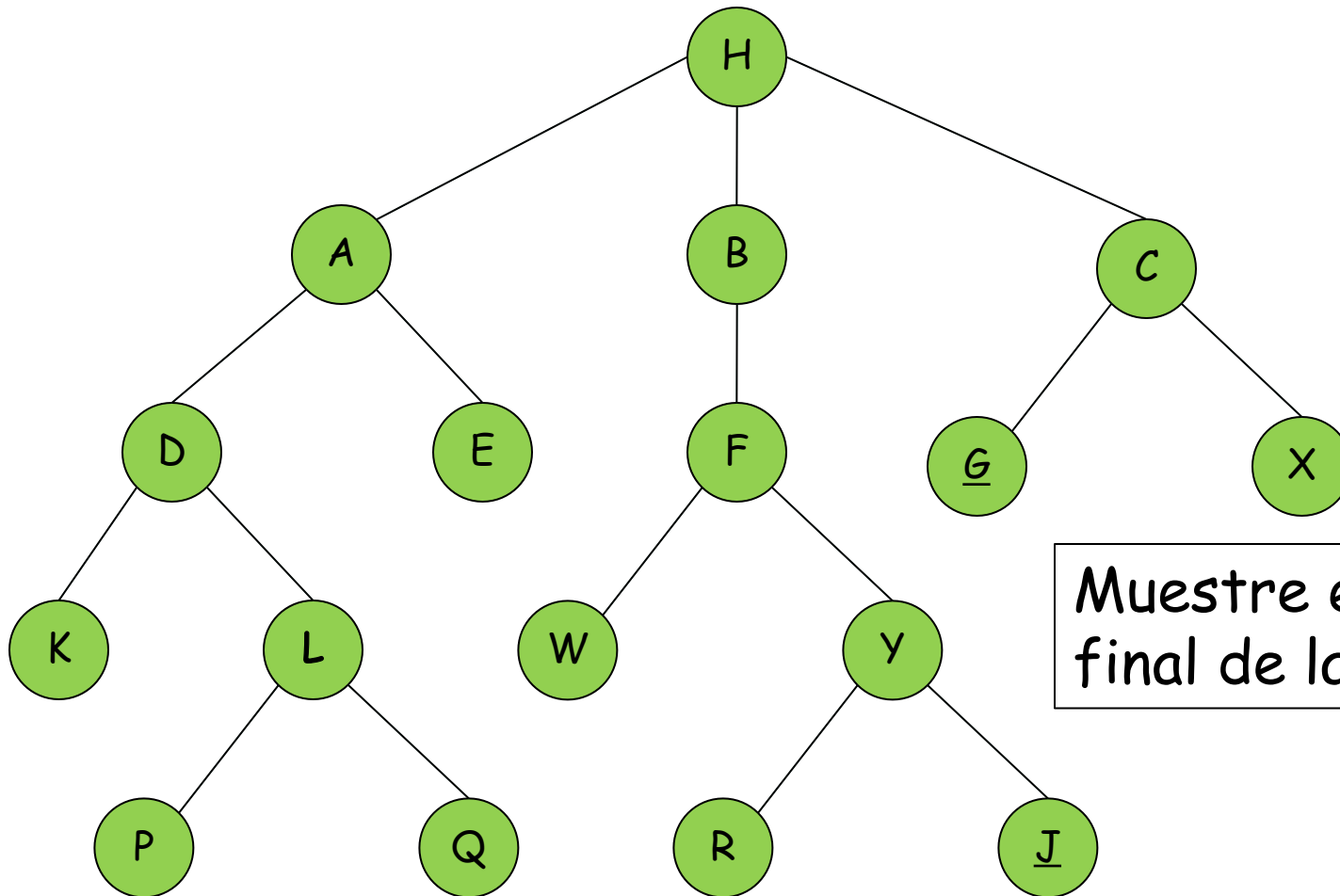
 expandir n y hacer push de todos sus hijos

Estado de la pila											Iteración
H											
	A	B	C								Extraer H, push hijos de H
		D	E	B	C						Extraer A, push hijos de A
			K	L	E	B	C				Extraer K
				L	E	B	C				Extraer L



Búsqueda no informada

Búsqueda preferente por profundidad



Muestre el estado final de la pila

Búsqueda no informada

Búsqueda preferente por profundidad

- Completitud
- Complejidad temporal
- Complejidad espacial
- Solución óptima

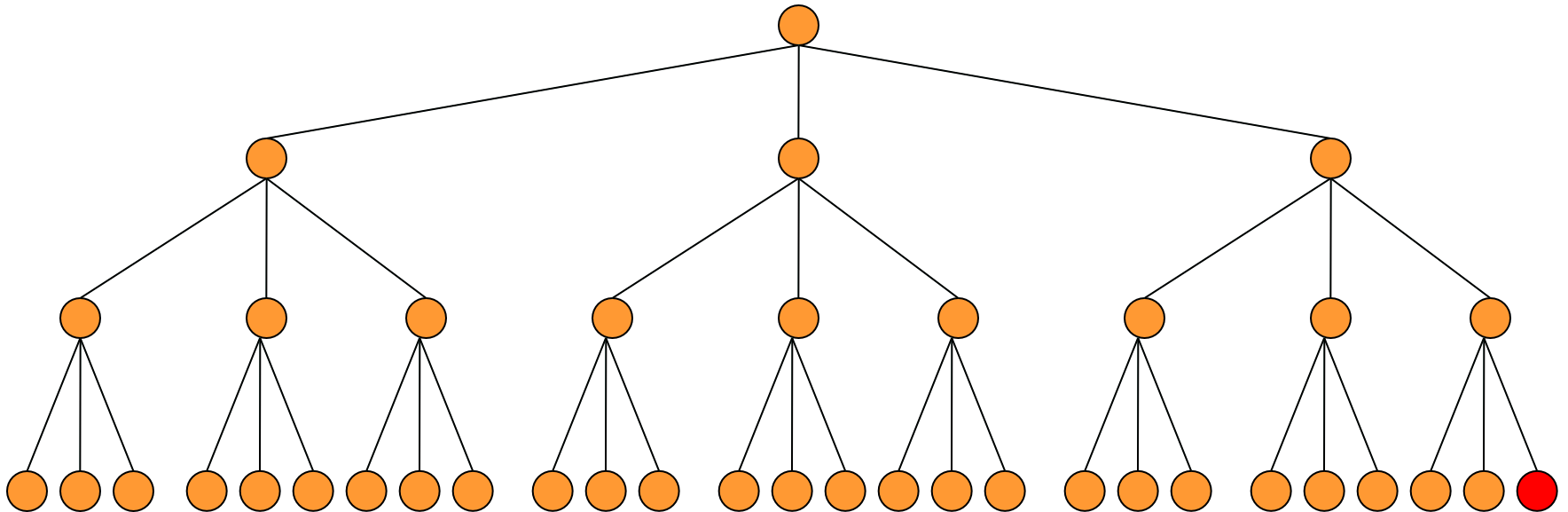
Búsqueda no informada

Búsqueda preferente por profundidad

- Completitud: no, puede quedar en un bucle infinito
- Complejidad temporal
- Complejidad espacial
- Solución óptima

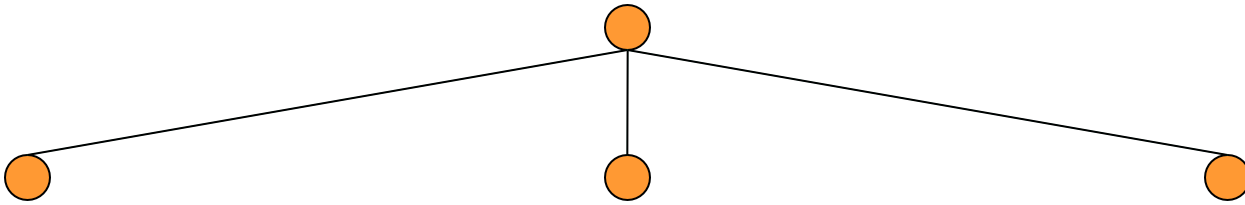
Búsqueda no informada

Búsqueda por profundidad



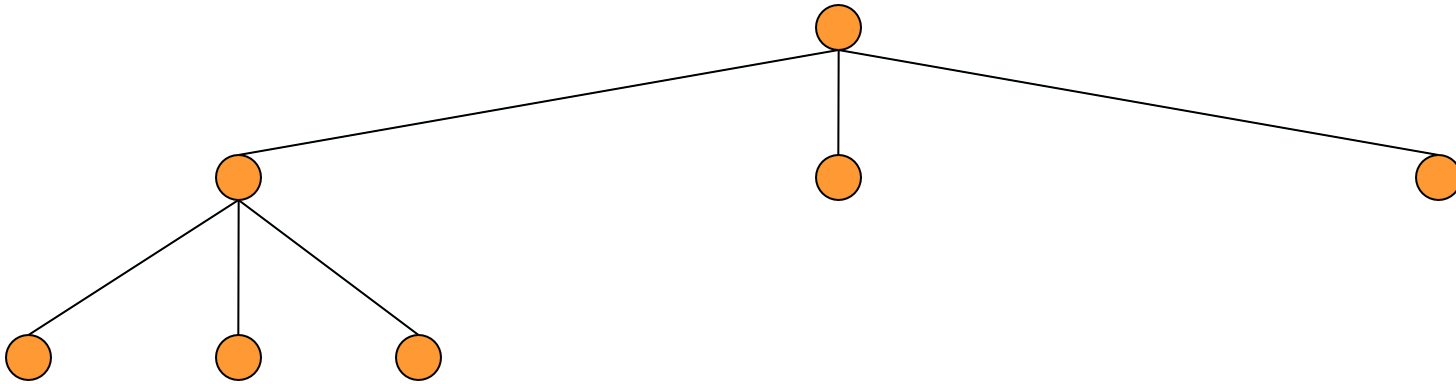
Búsqueda no informada

Búsqueda por profundidad



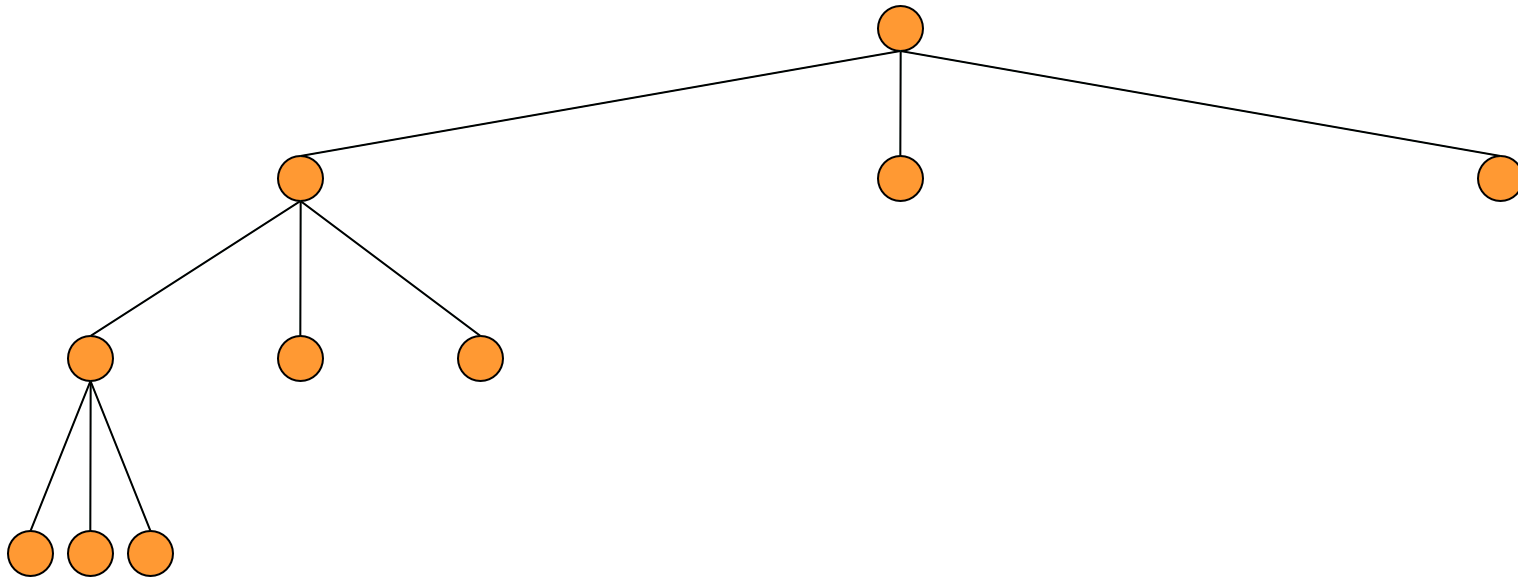
Búsqueda no informada

Búsqueda por profundidad



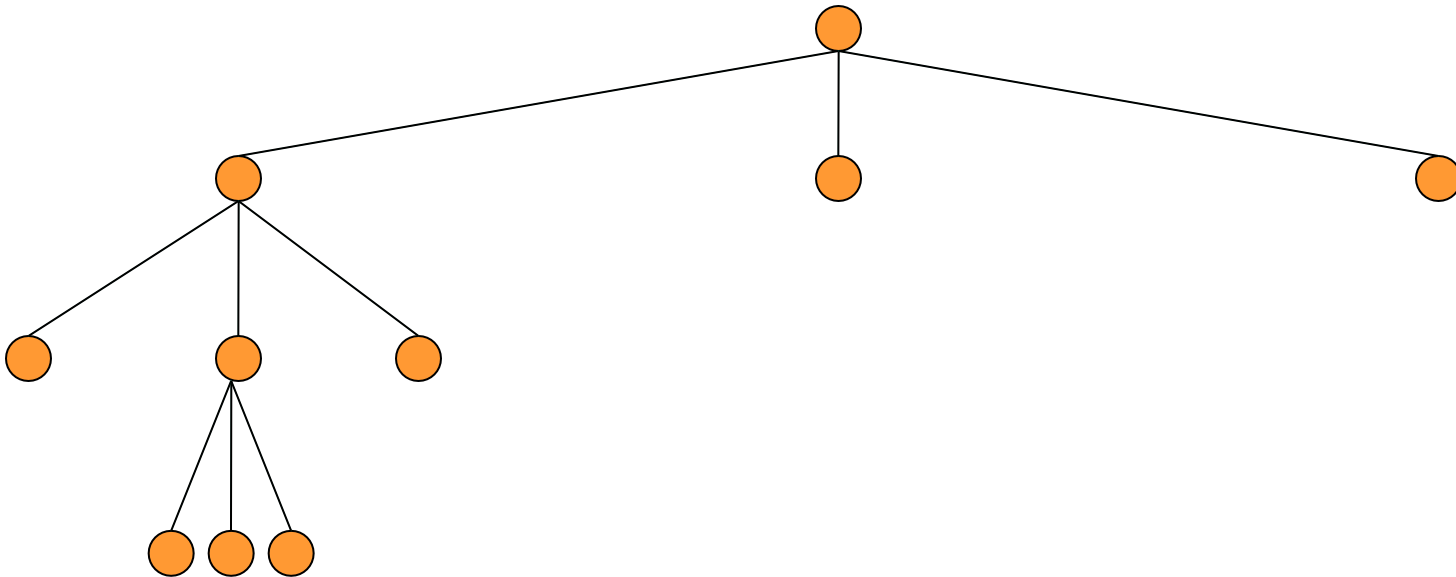
Búsqueda no informada

Búsqueda por profundidad



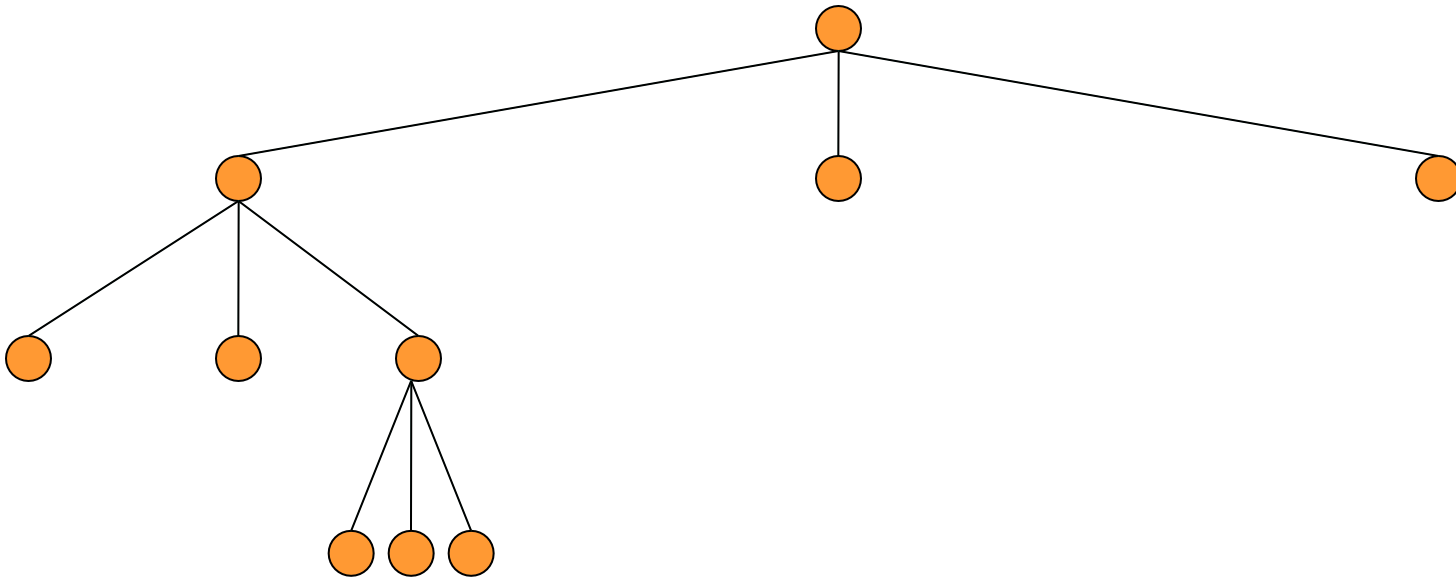
Búsqueda no informada

Búsqueda por profundidad



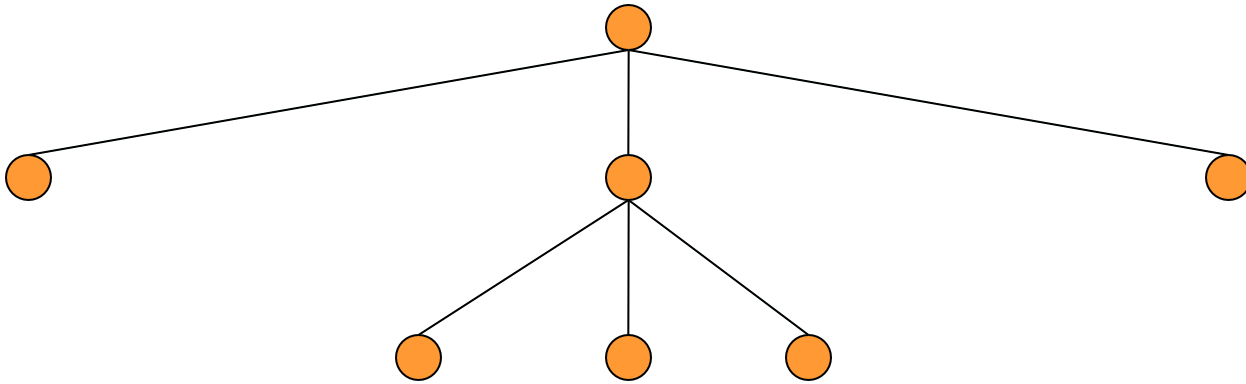
Búsqueda no informada

Búsqueda por profundidad



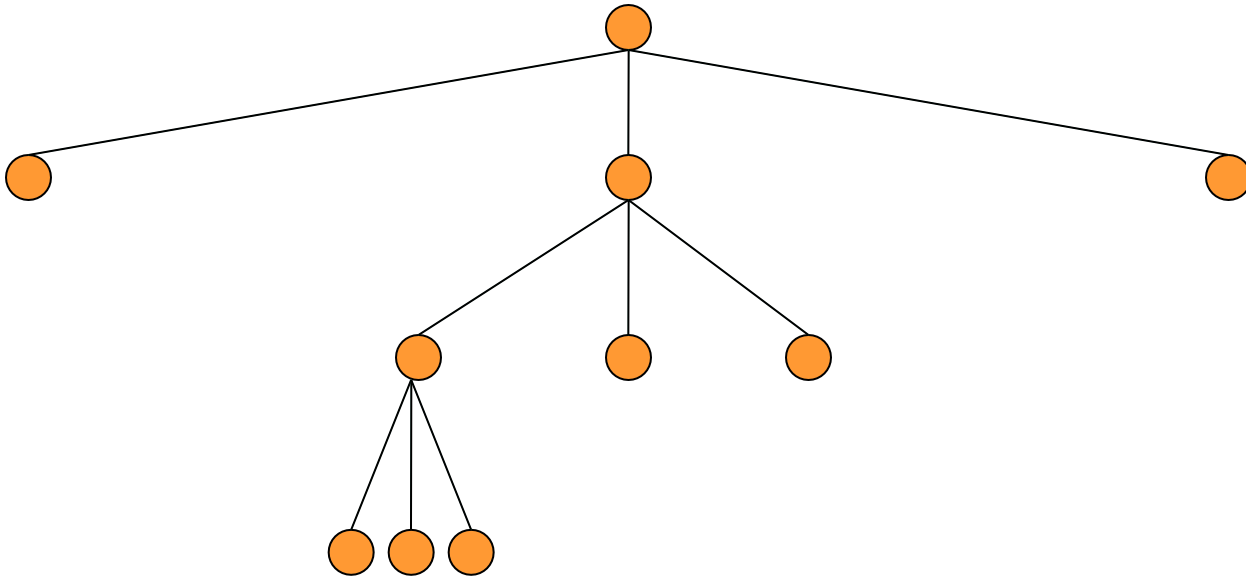
Búsqueda no informada

Búsqueda por profundidad



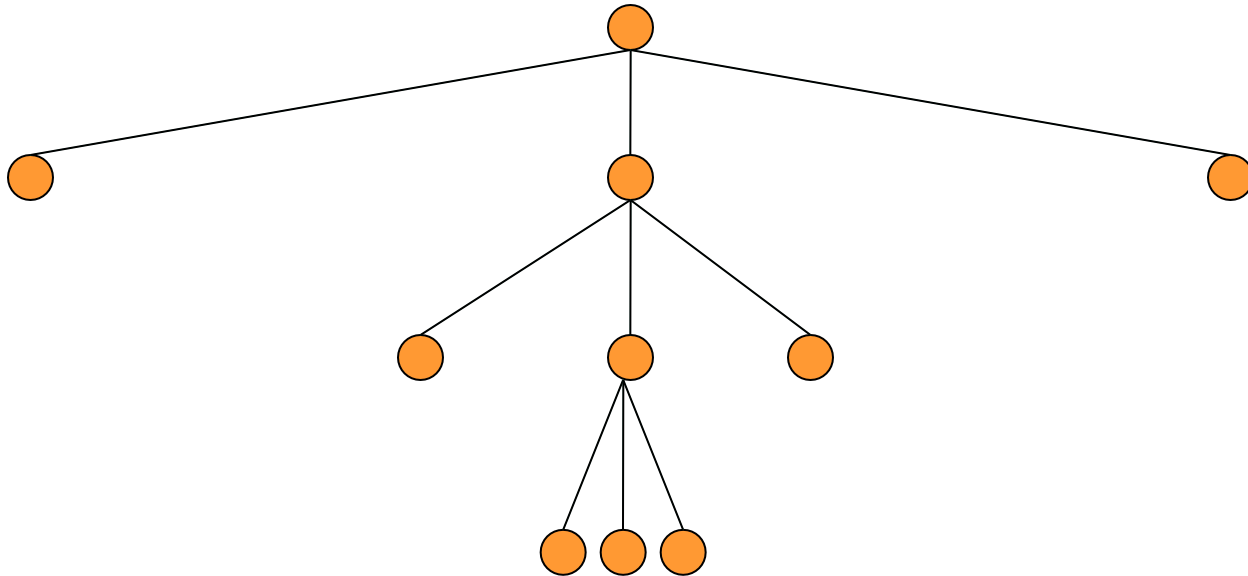
Búsqueda no informada

Búsqueda por profundidad



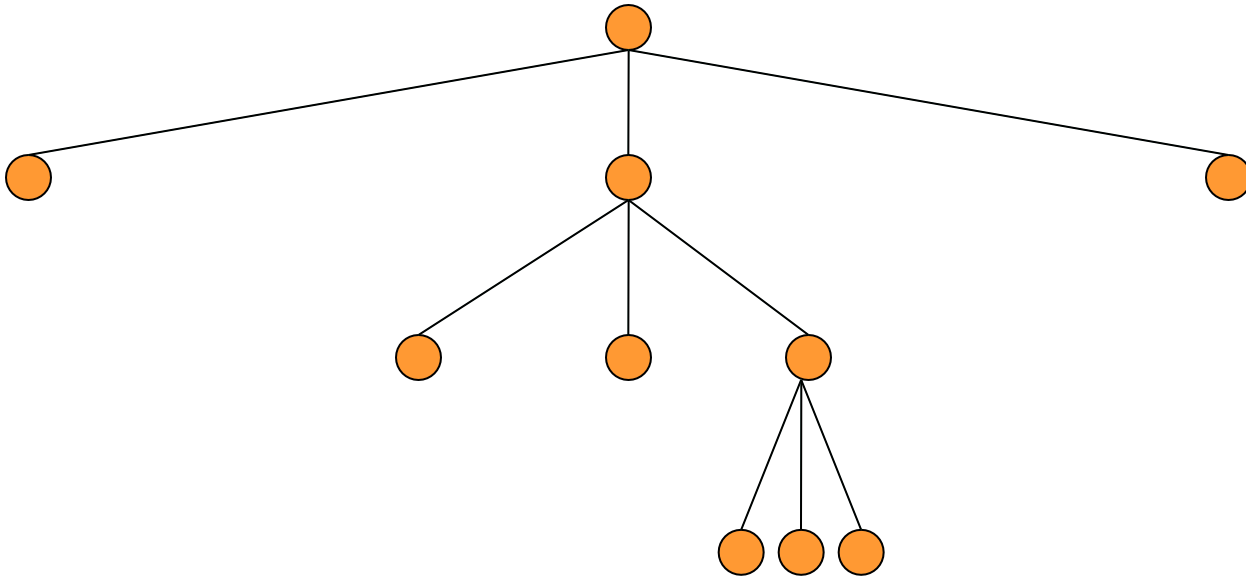
Búsqueda no informada

Búsqueda por profundidad



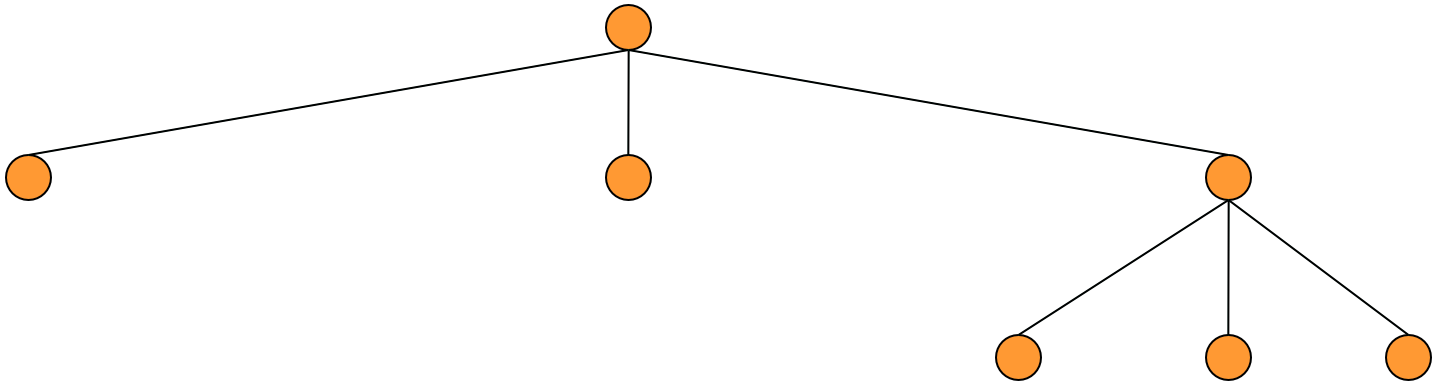
Búsqueda no informada

Búsqueda por profundidad



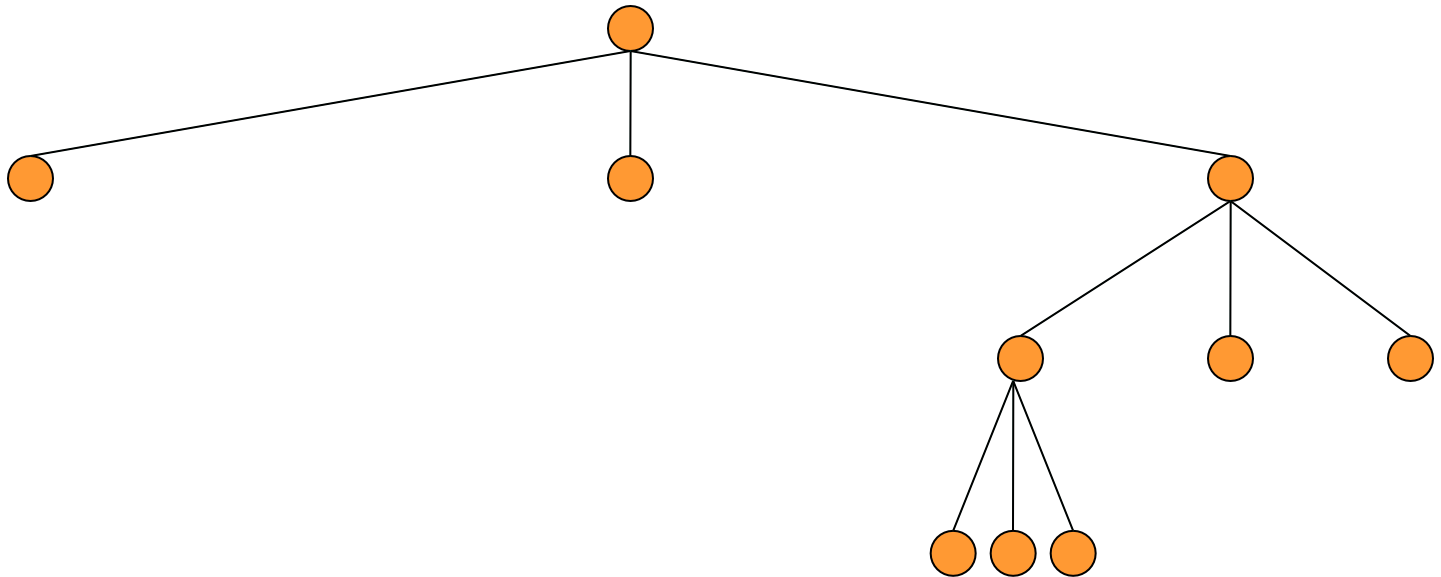
Búsqueda no informada

Búsqueda por profundidad



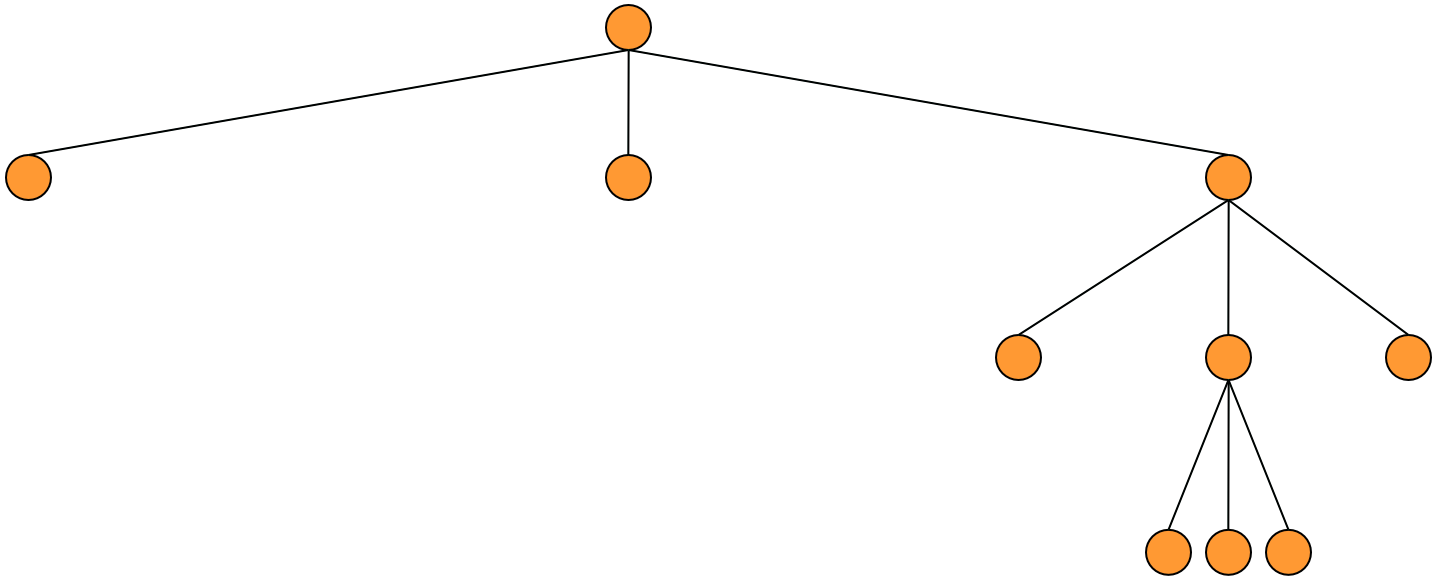
Búsqueda no informada

Búsqueda por profundidad



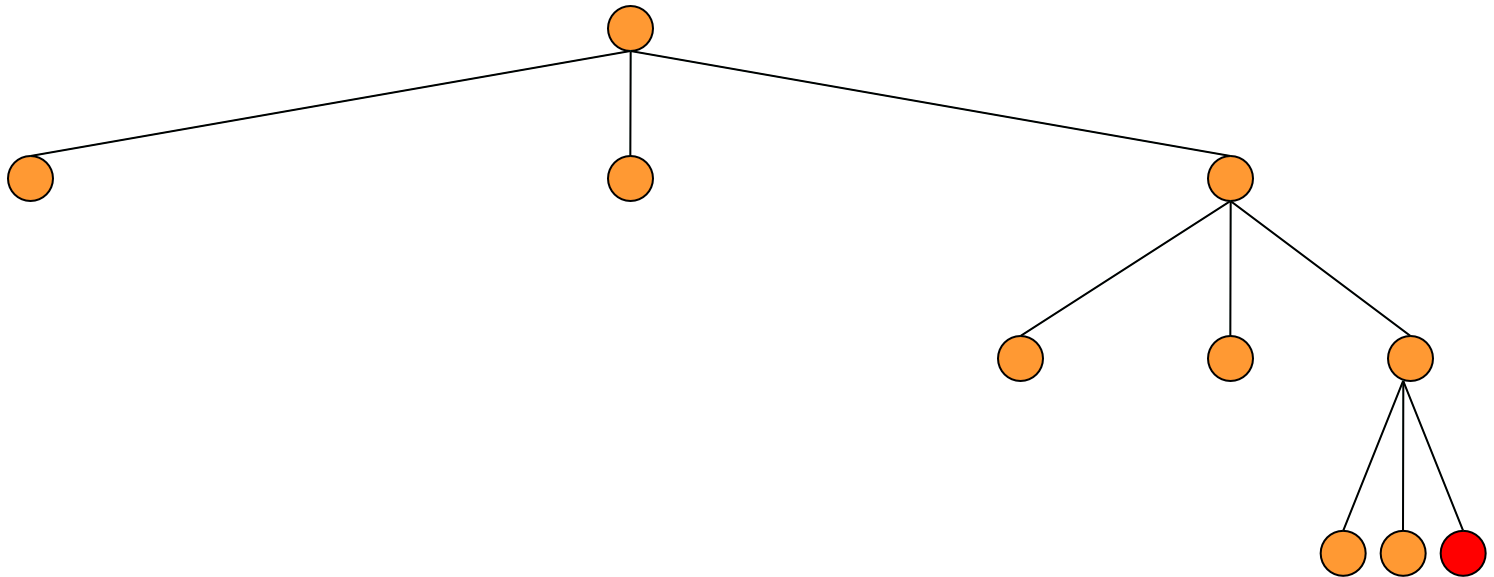
Búsqueda no informada

Búsqueda por profundidad



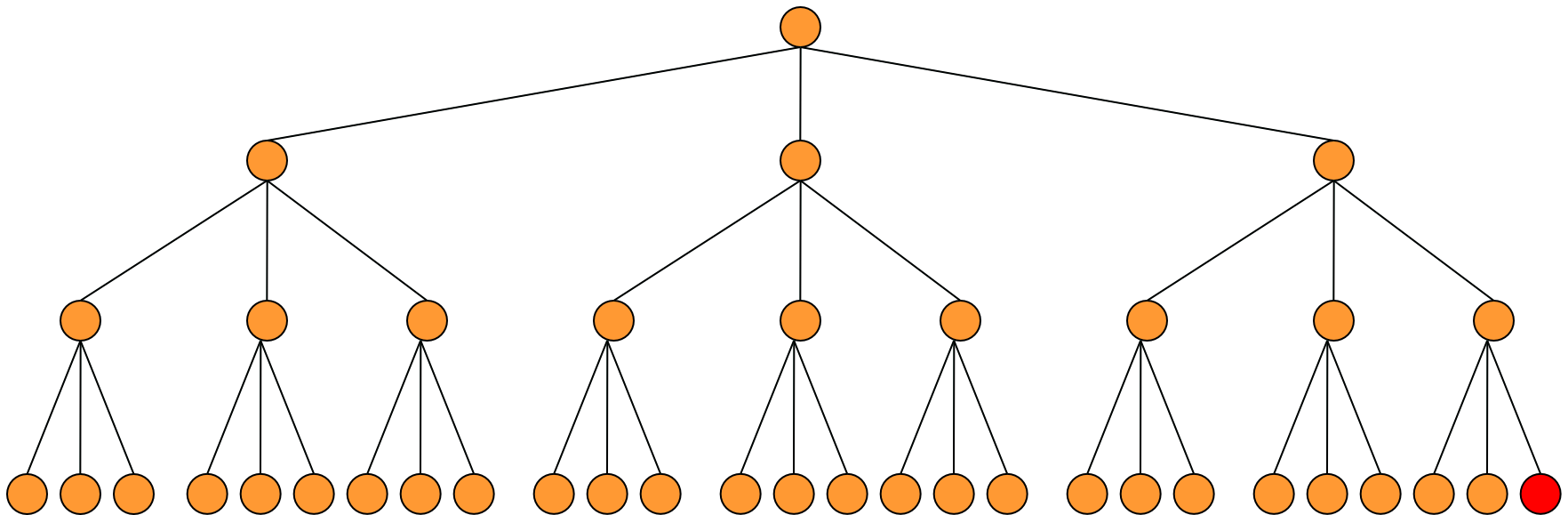
Búsqueda no informada

Búsqueda por profundidad



Búsqueda no informada

Búsqueda por profundidad



¿En total cuántos nodos se expanden?

¿Cuántos nodos se deben almacenar?

Búsqueda no informada

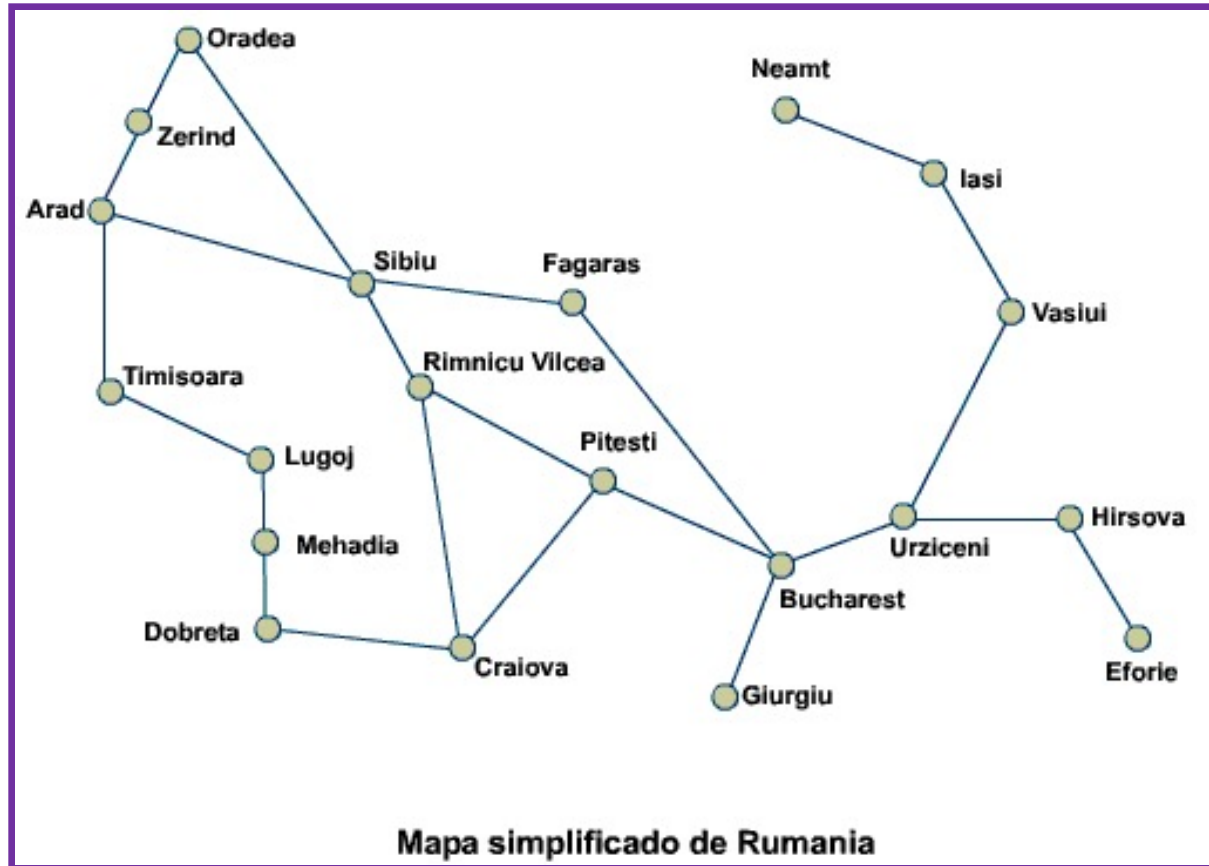
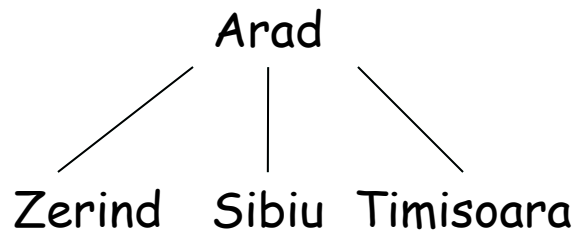
Búsqueda preferente por profundidad

- **Complejidad:** no, puede quedar en un bucle infinito
- **Complejidad temporal:** $O(b^d)$, en el peor caso se consideran los mismos nodos que por amplitud, simplemente que en un orden distinto
- **Complejidad espacial:** no es necesario guardar cada nodo del árbol en memoria. Complejidad $b \cdot d$
- **Solución óptima:** no

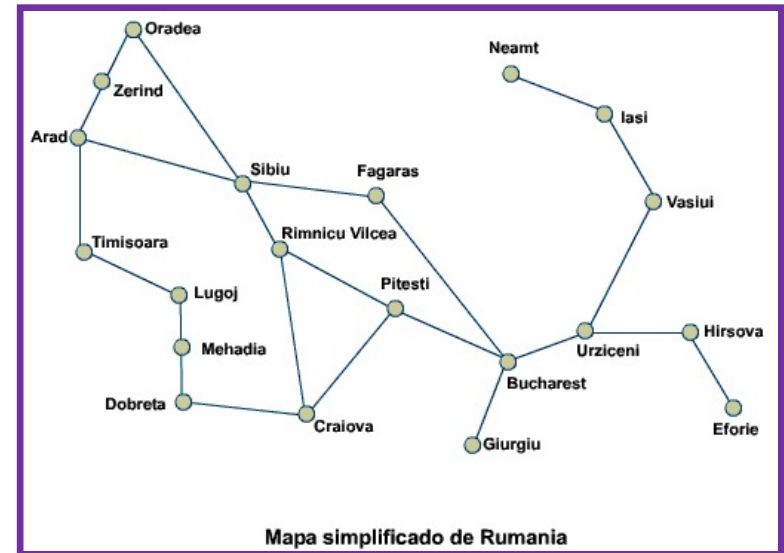
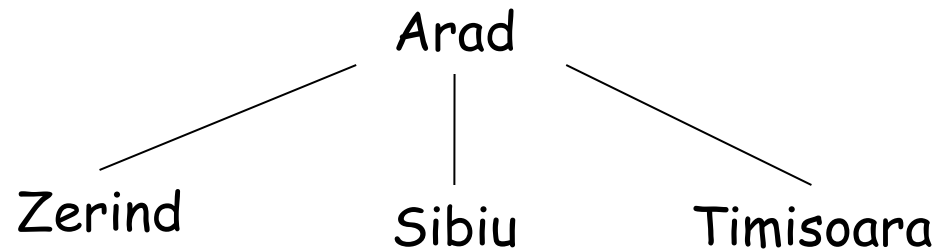
Búsqueda no informada

Aplicar búsqueda
preferente por
profundidad para ir de
Arad a Bucarest

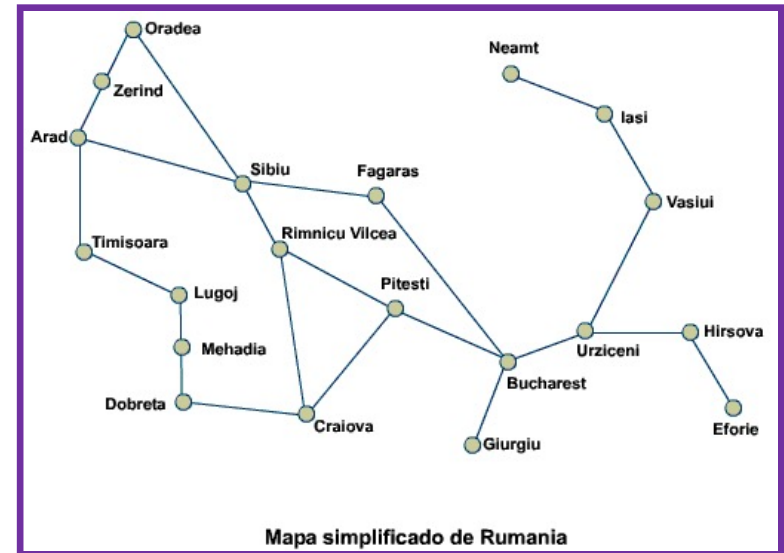
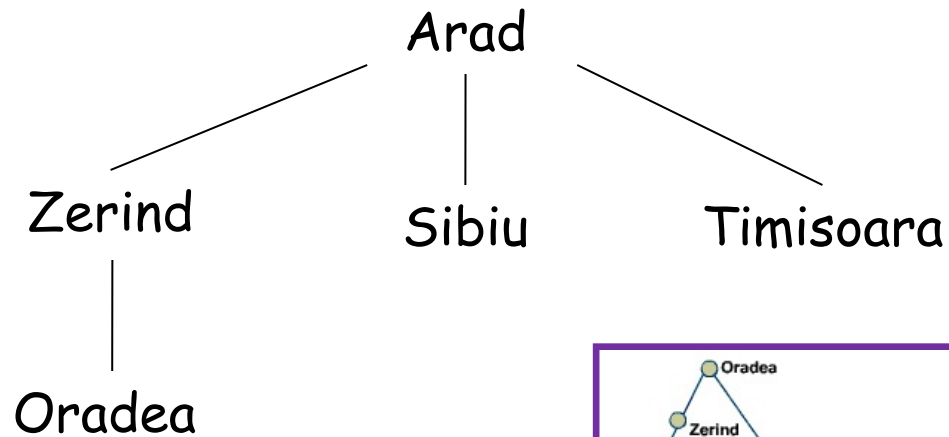
* Evite devolverse



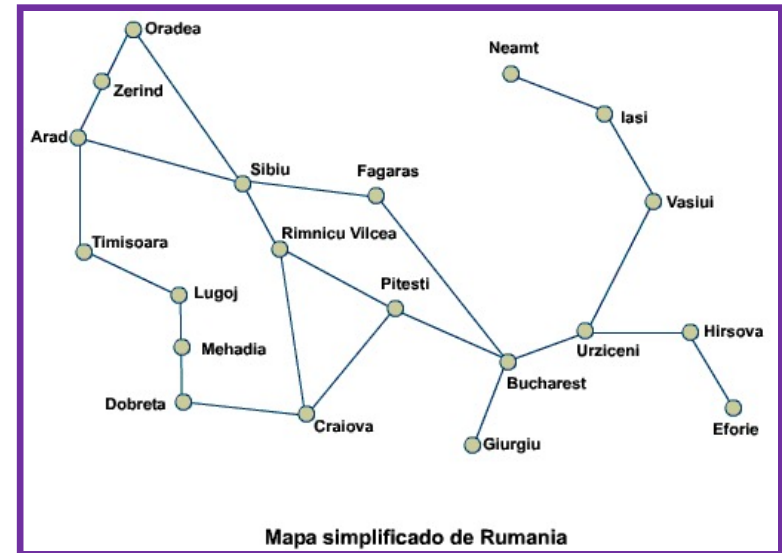
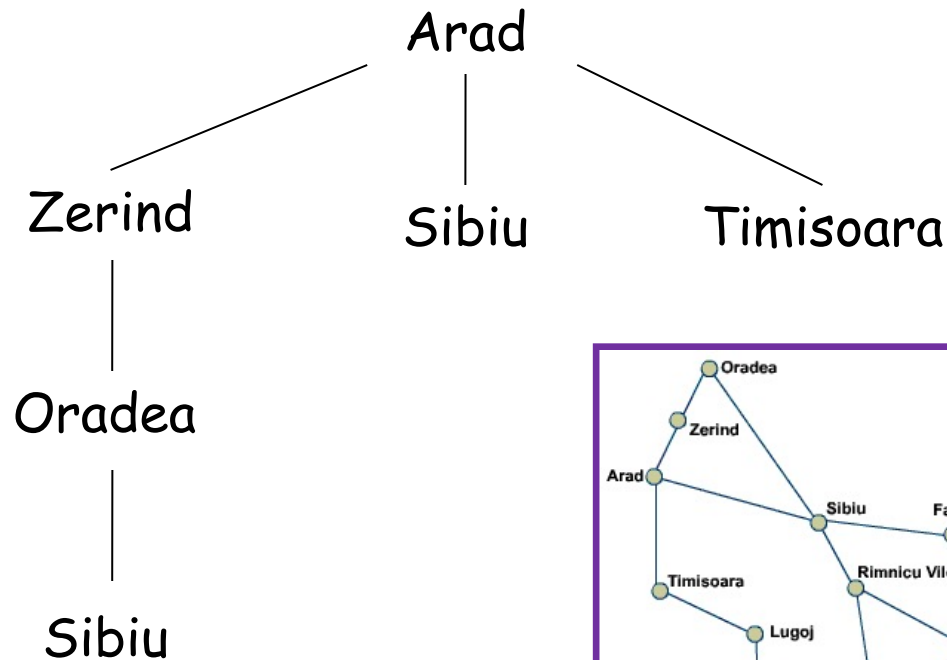
Búsqueda no informada



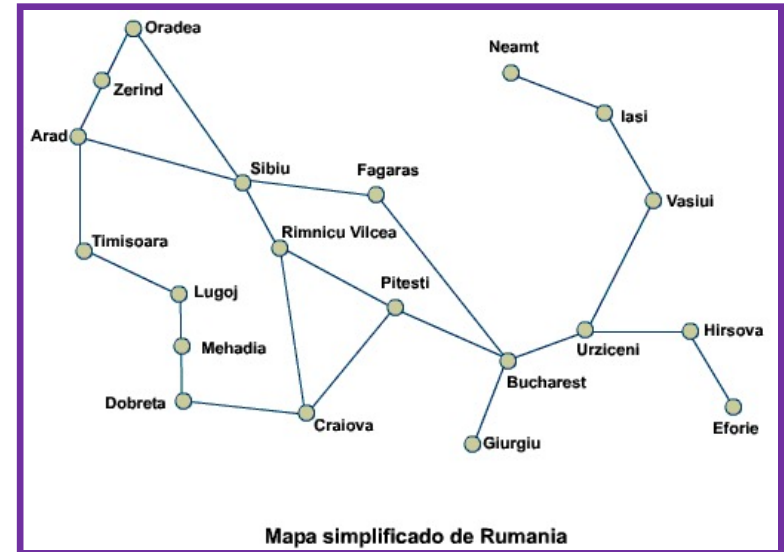
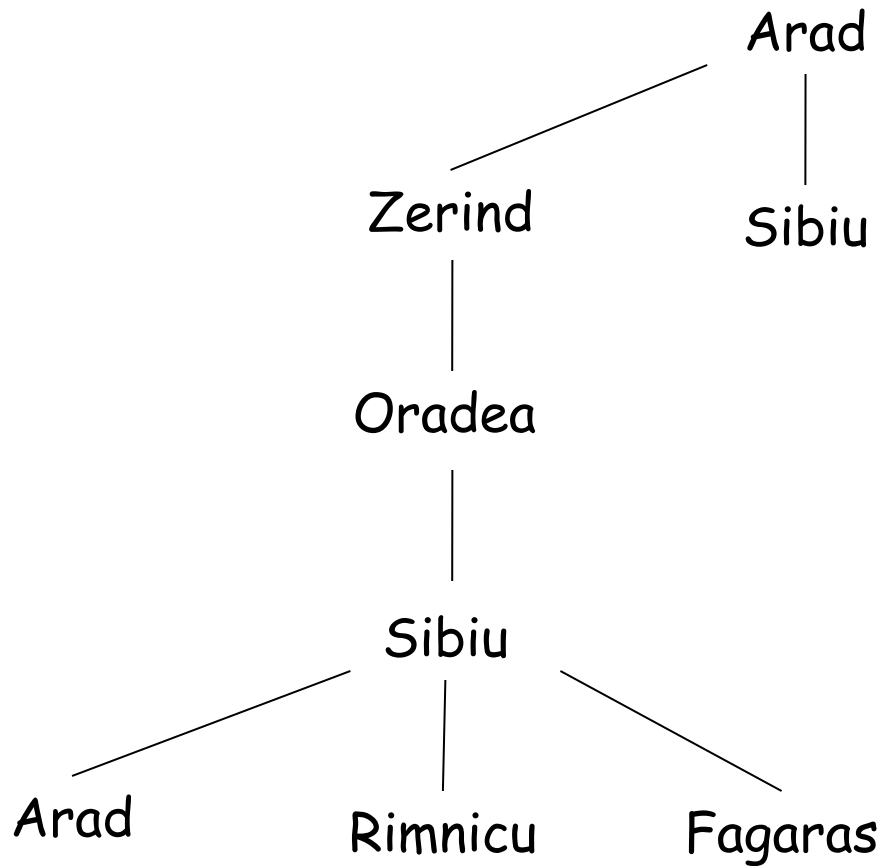
Búsqueda no informada



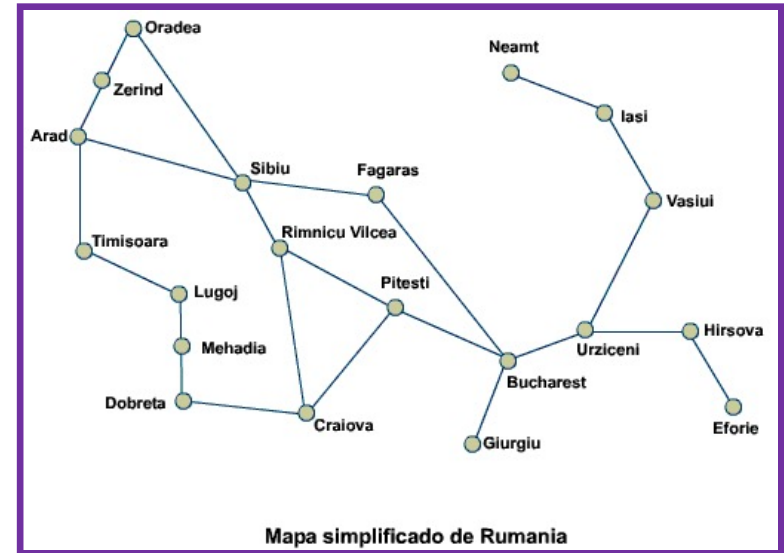
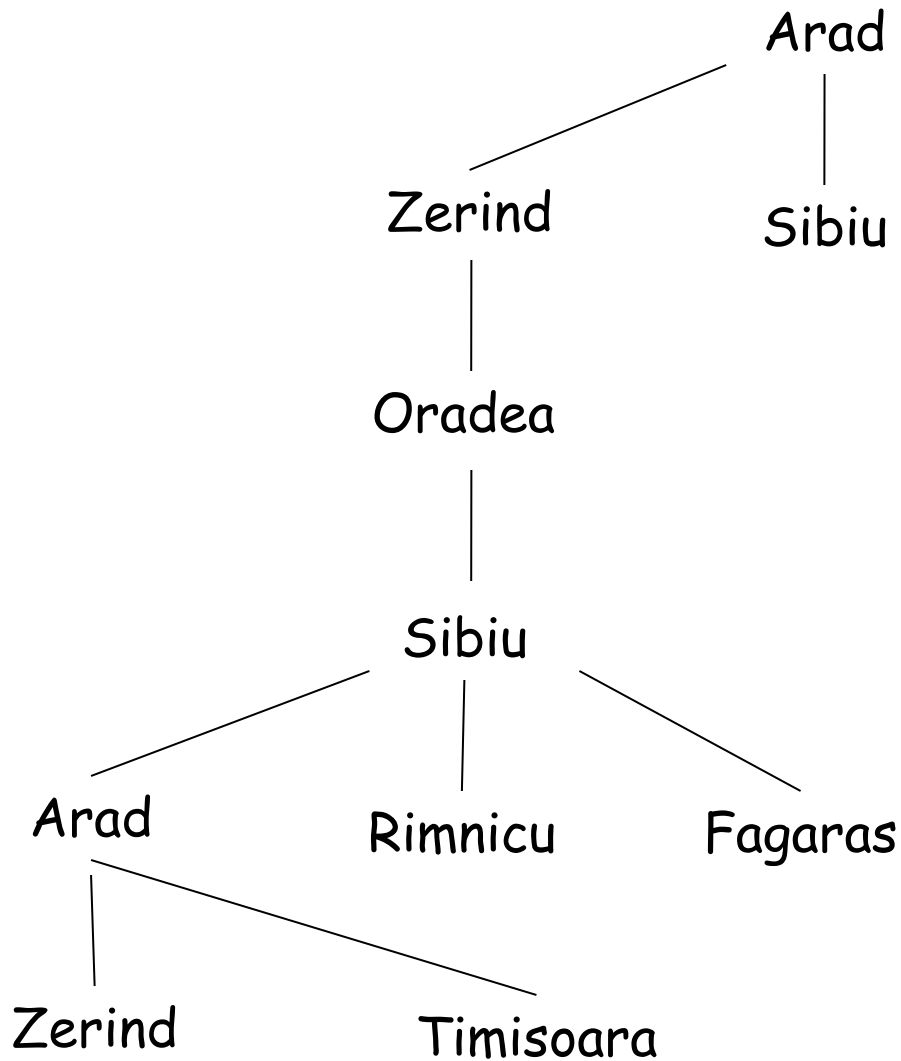
Búsqueda no informada



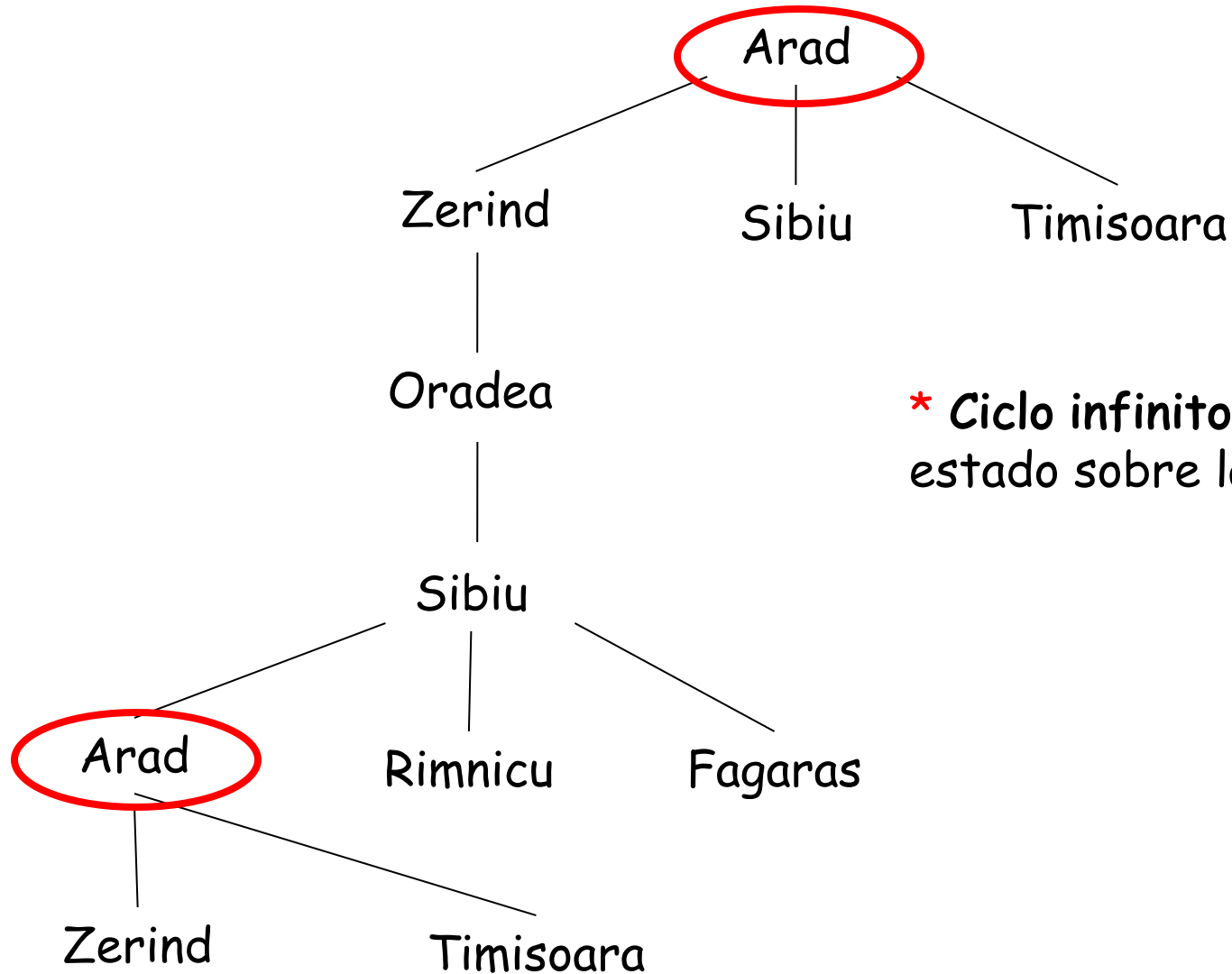
Búsqueda no informada



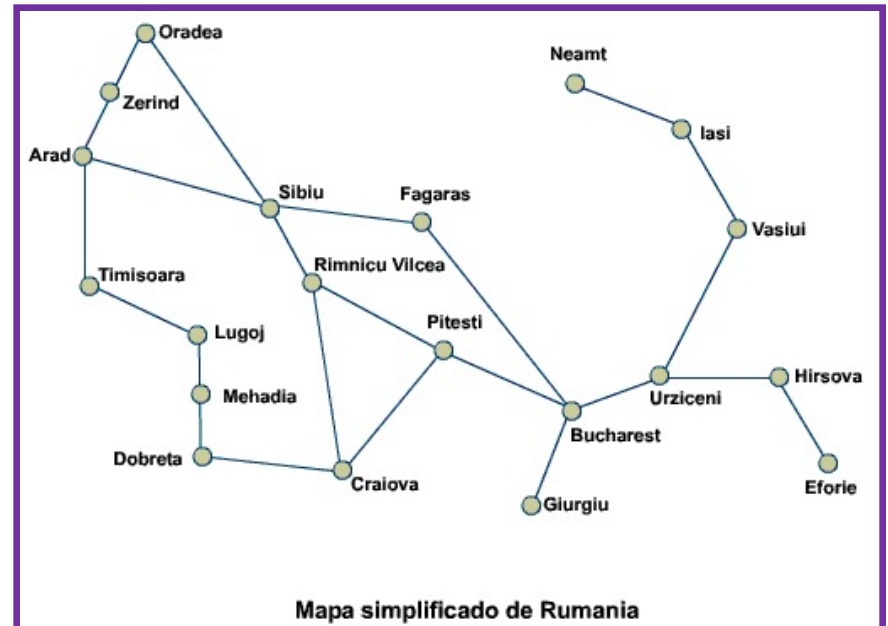
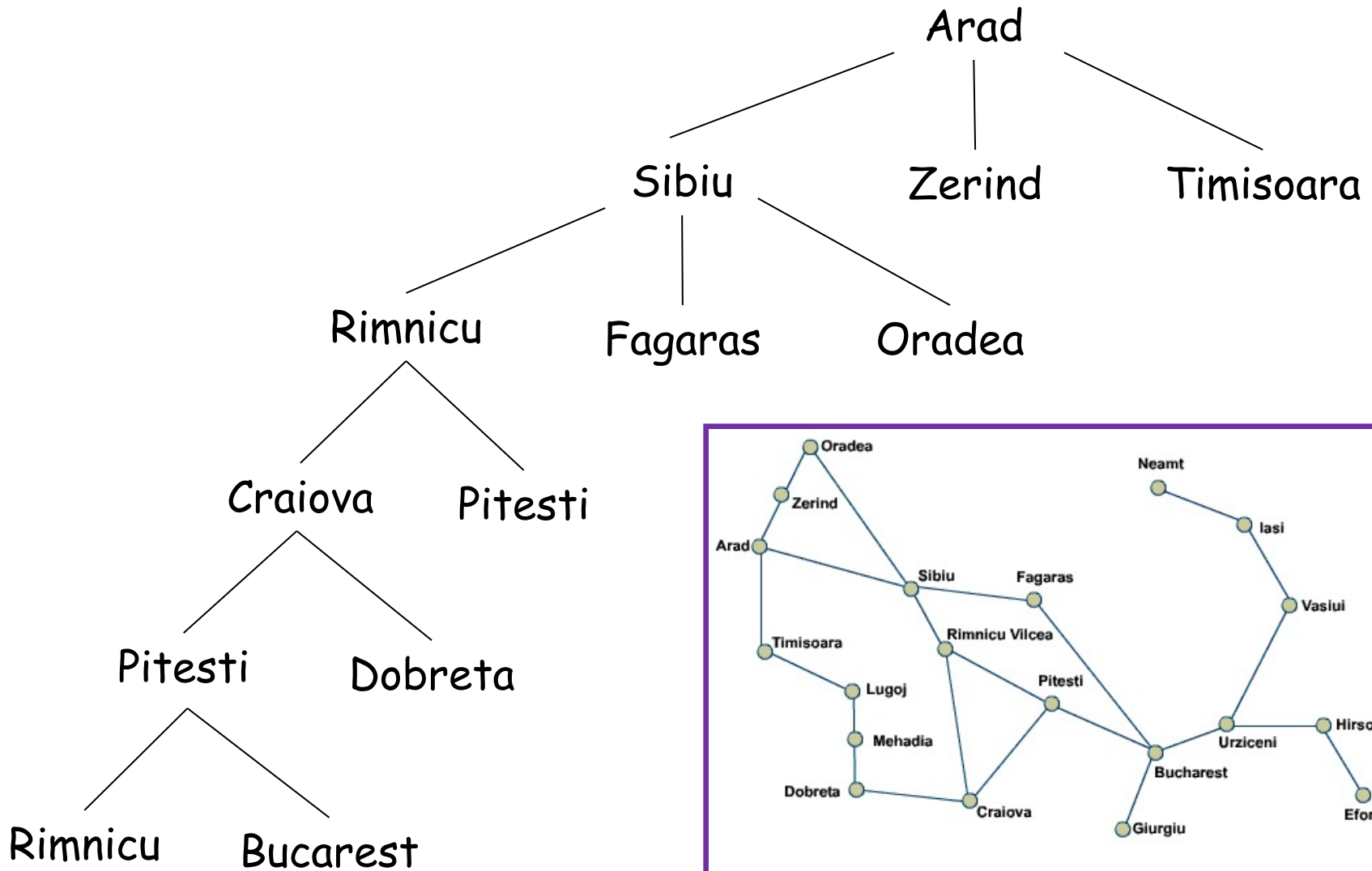
Búsqueda no informada



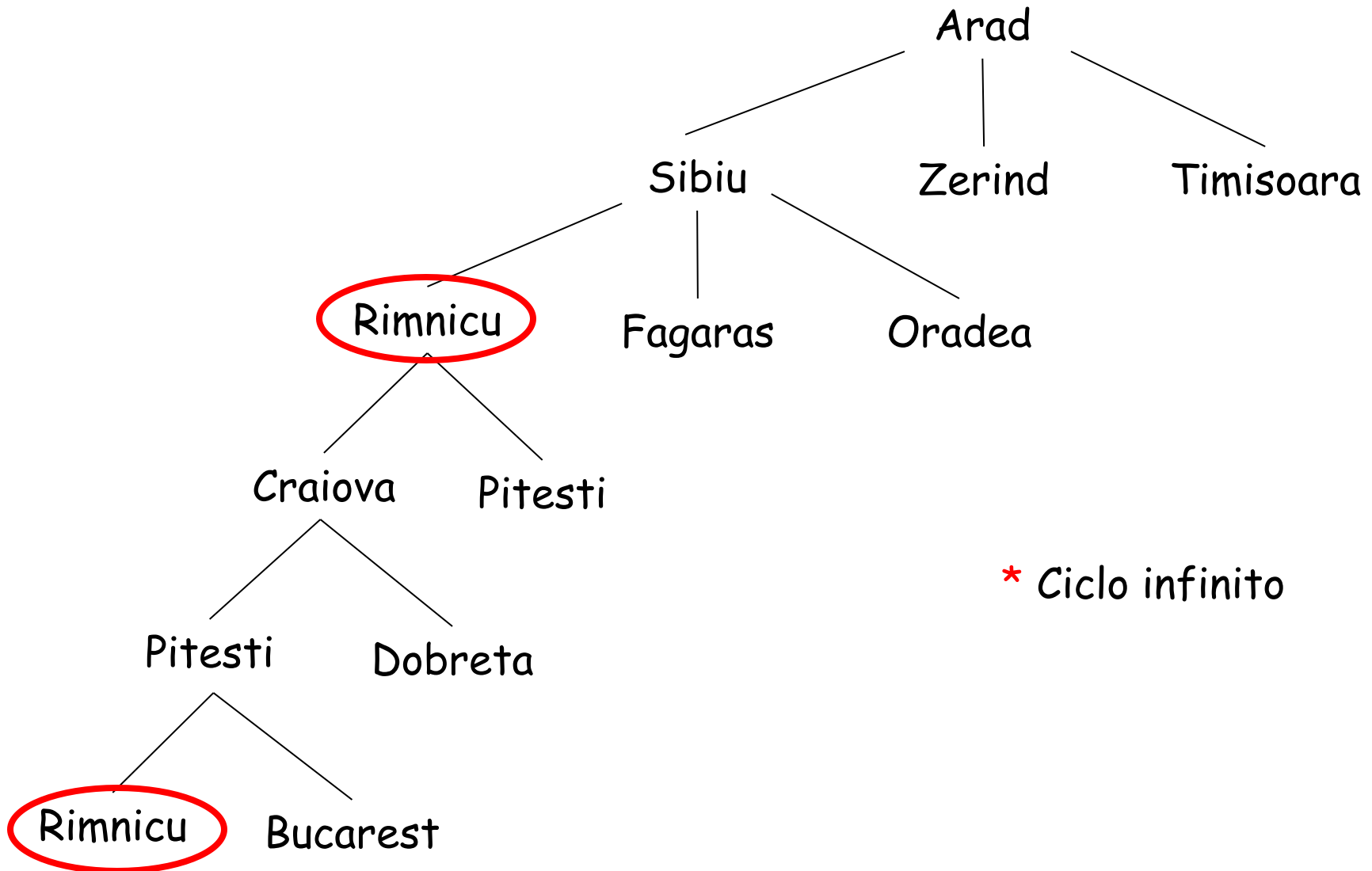
Búsqueda no informada



Búsqueda no informada



Búsqueda no informada



Búsqueda no informada

Búsqueda preferente por profundidad evitando ciclos

- Se evita que sobre una misma rama se alcance un estado previo

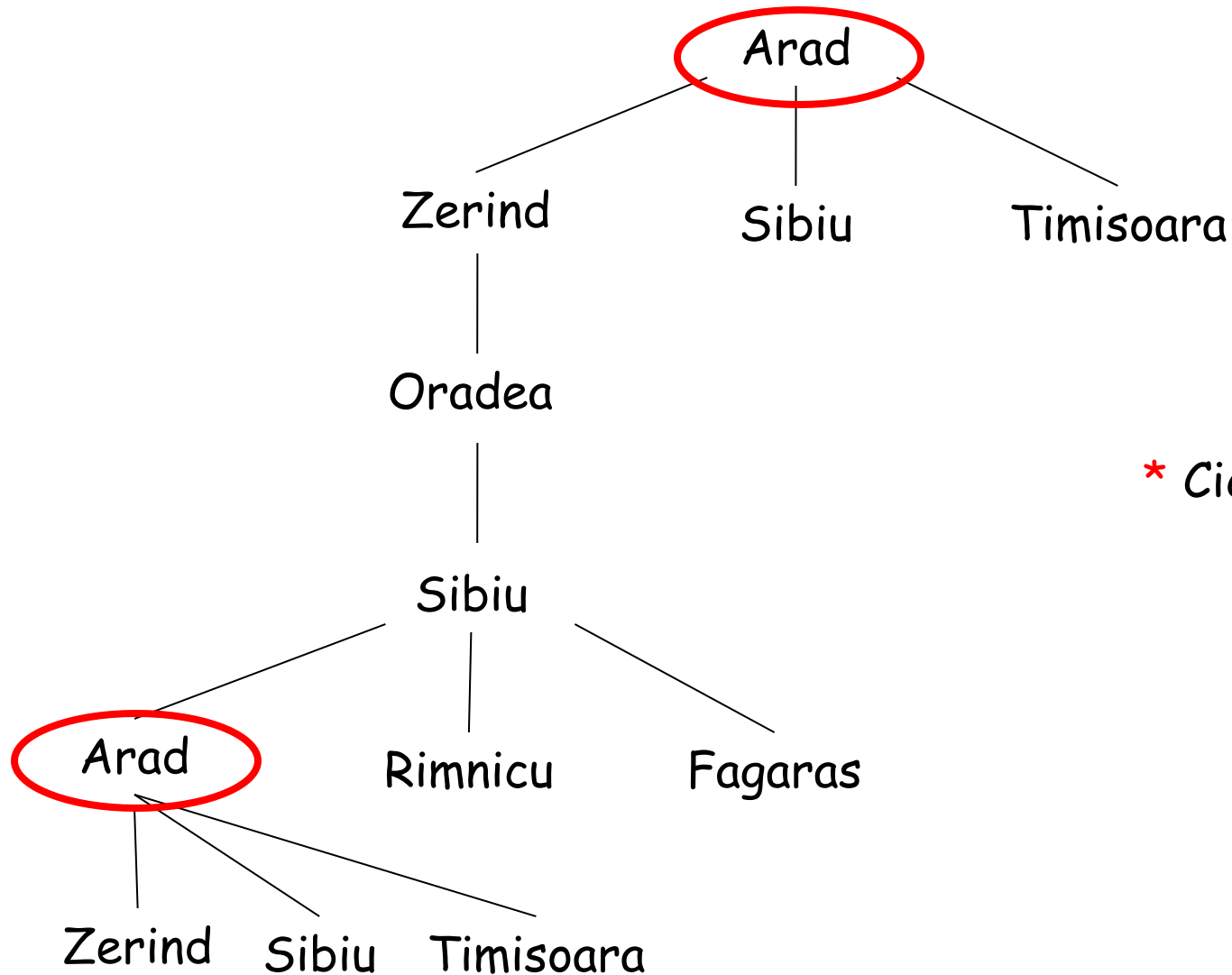
Búsqueda no informada

Búsqueda preferente por profundidad evitando ciclos

- Se evita que sobre una misma rama se alcance un estado previo

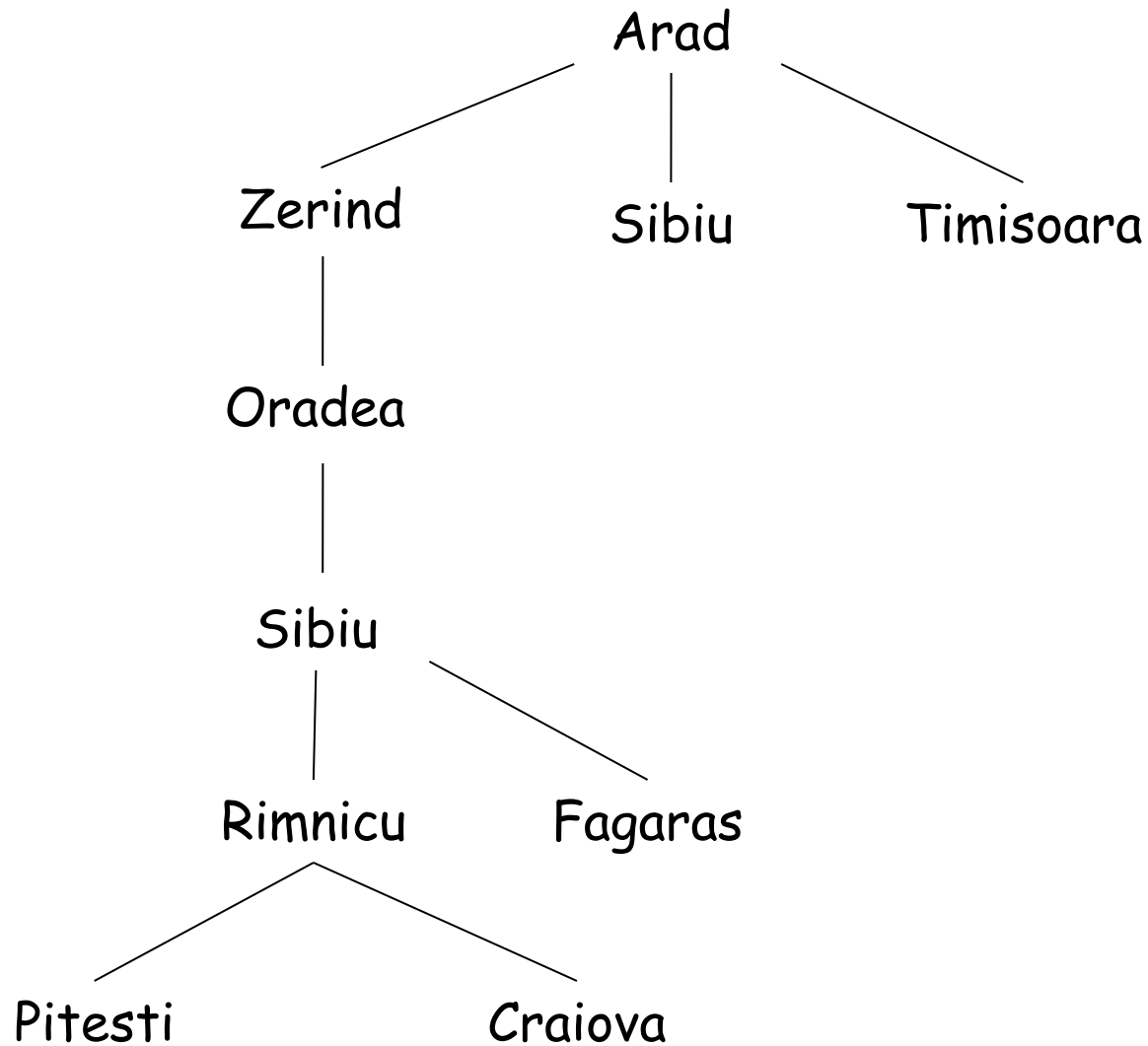
"Evitar ciclo" es un concepto más completo que "evitar devolverse"

Búsqueda no informada

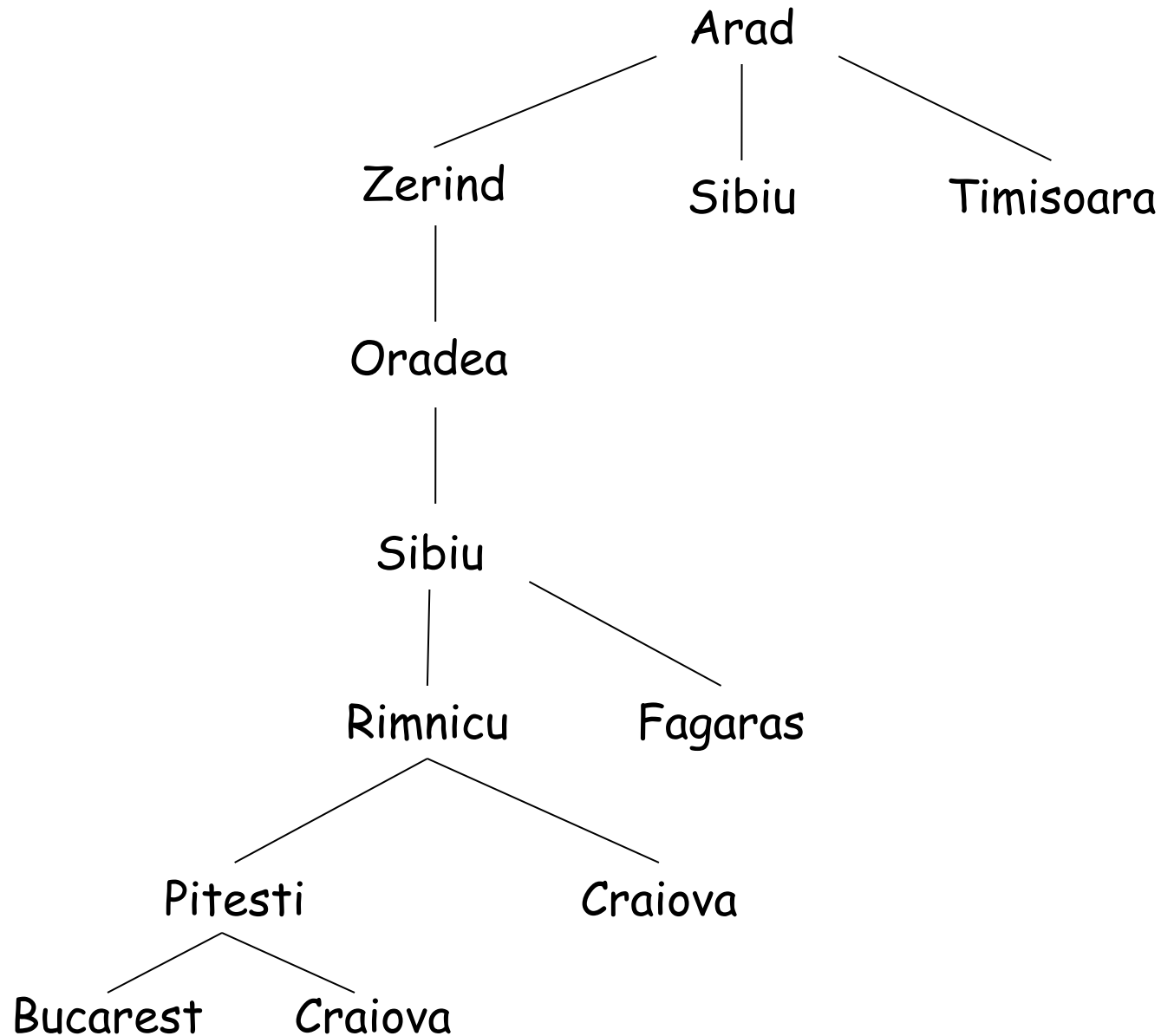


* Ciclo infinito

Búsqueda no informada

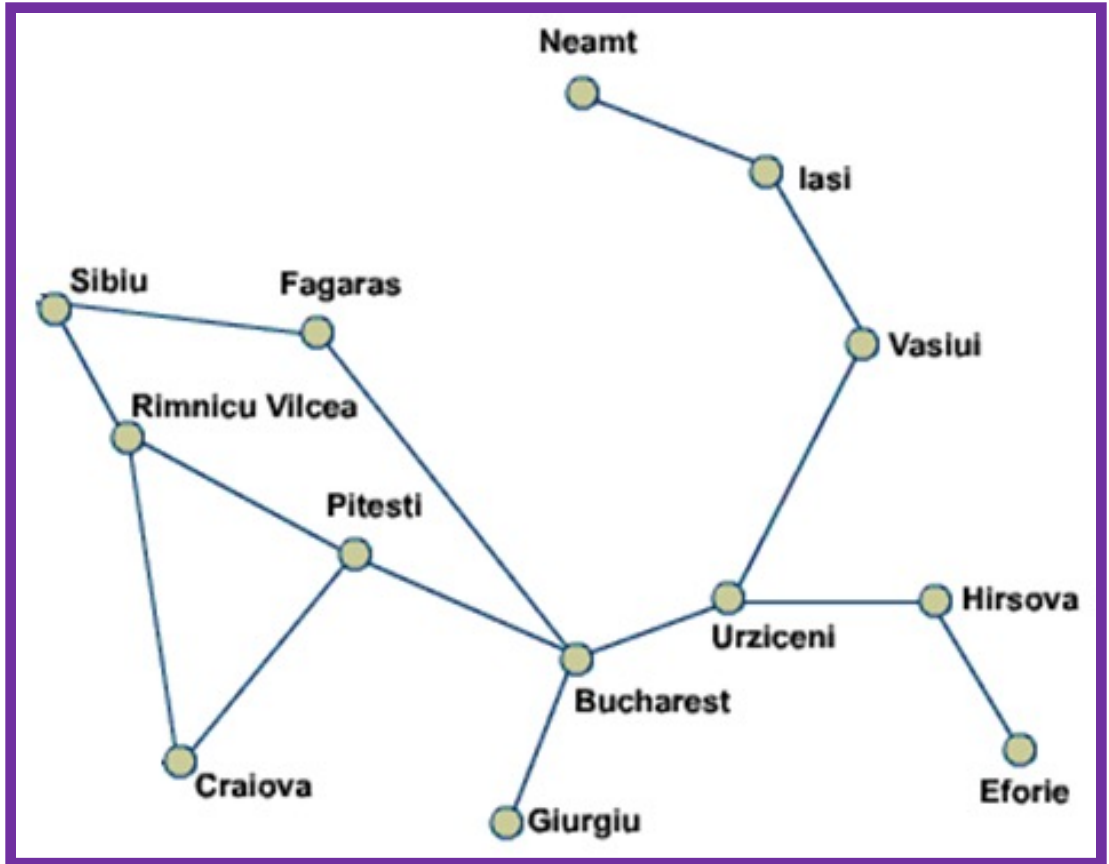
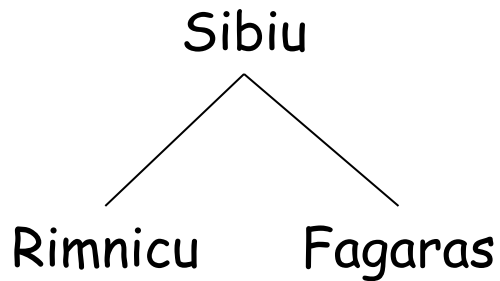


Búsqueda no informada

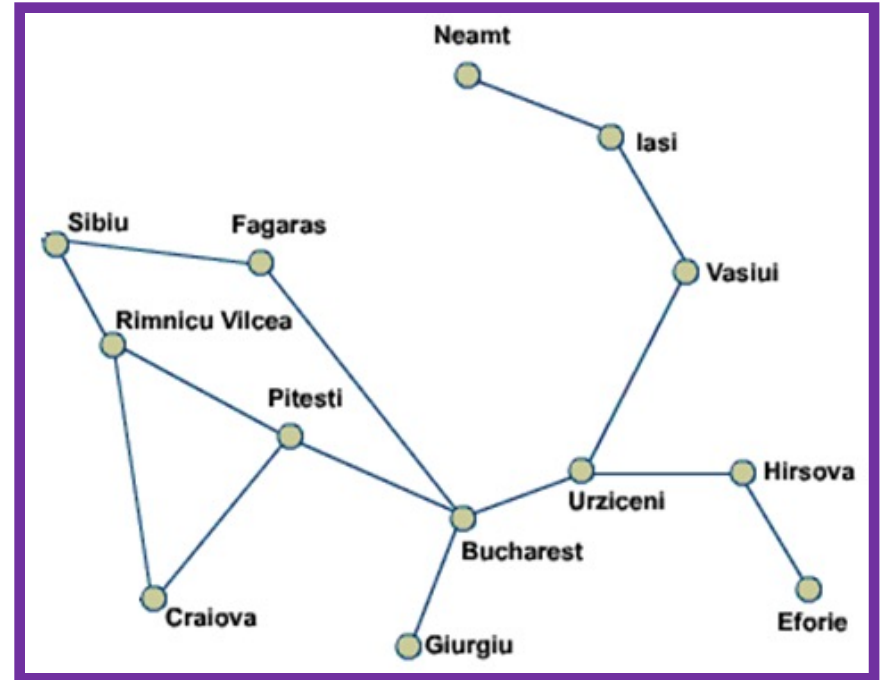
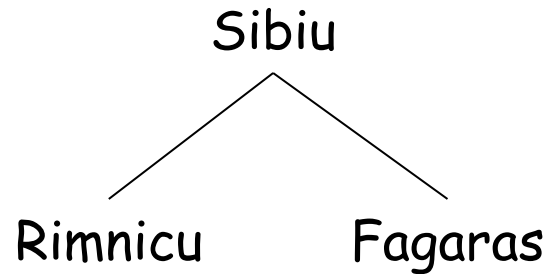


Búsqueda no informada

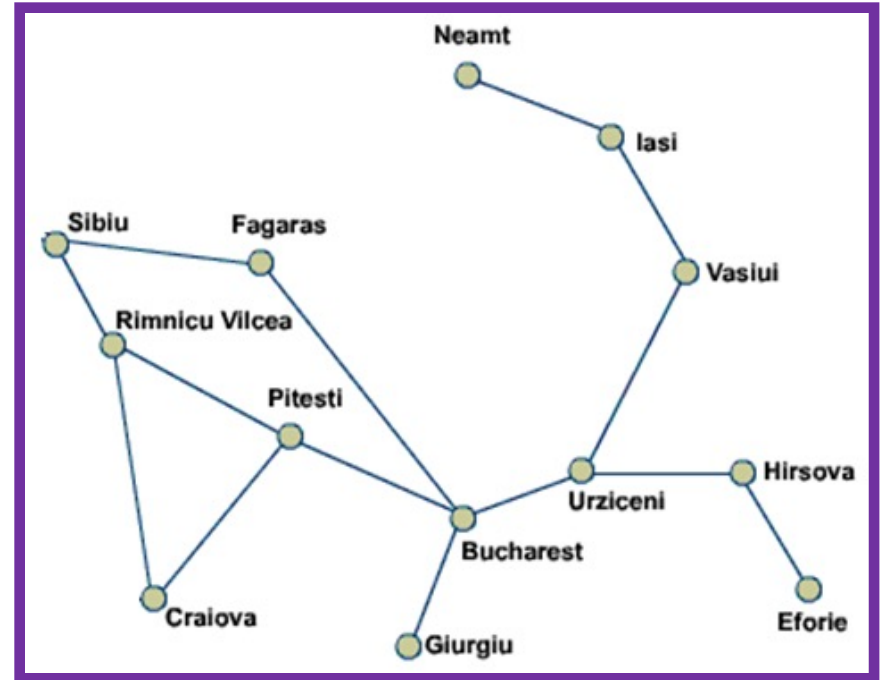
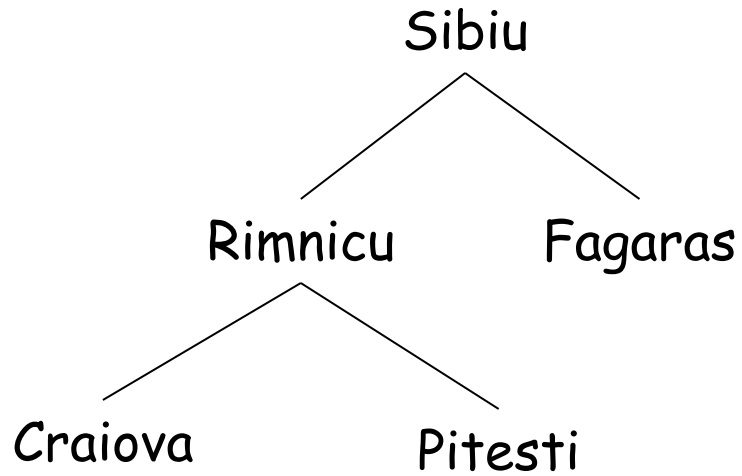
Aplicar búsqueda
preferente por profundidad
evitando ciclos



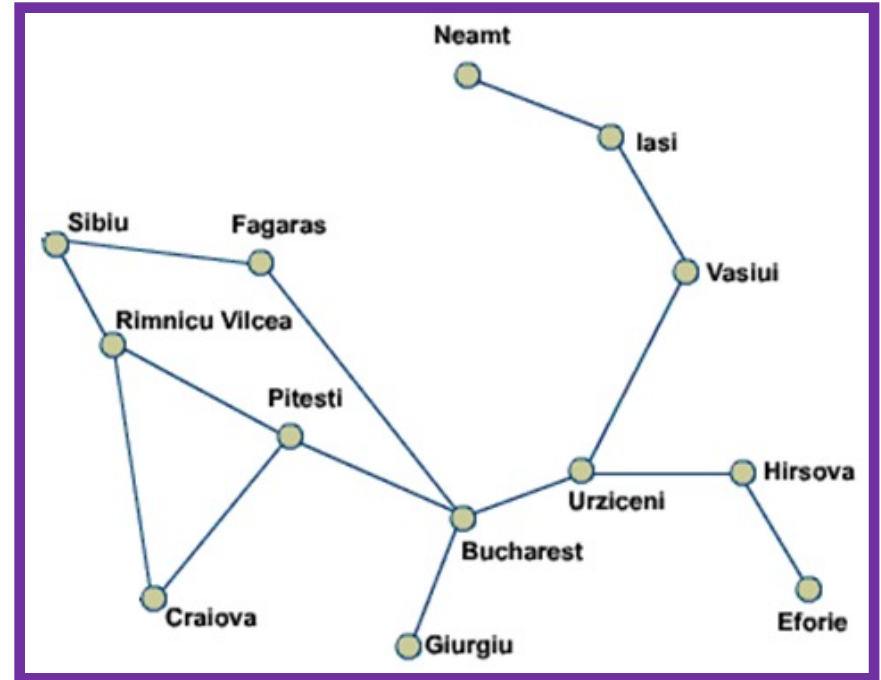
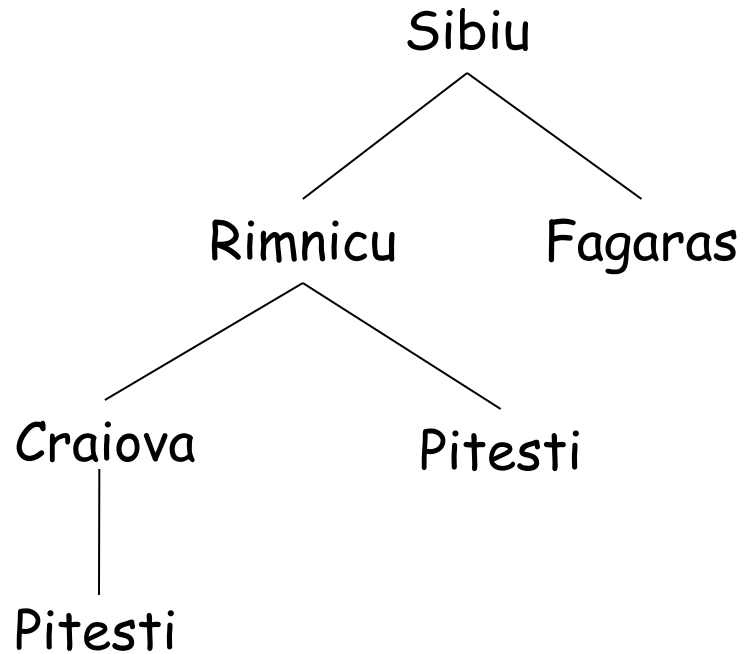
Búsqueda no informada



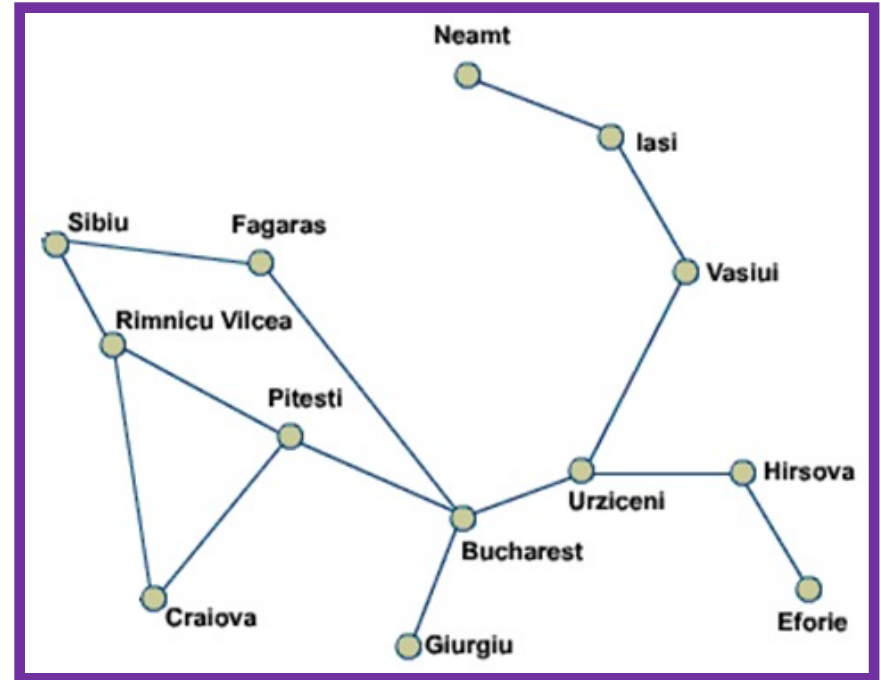
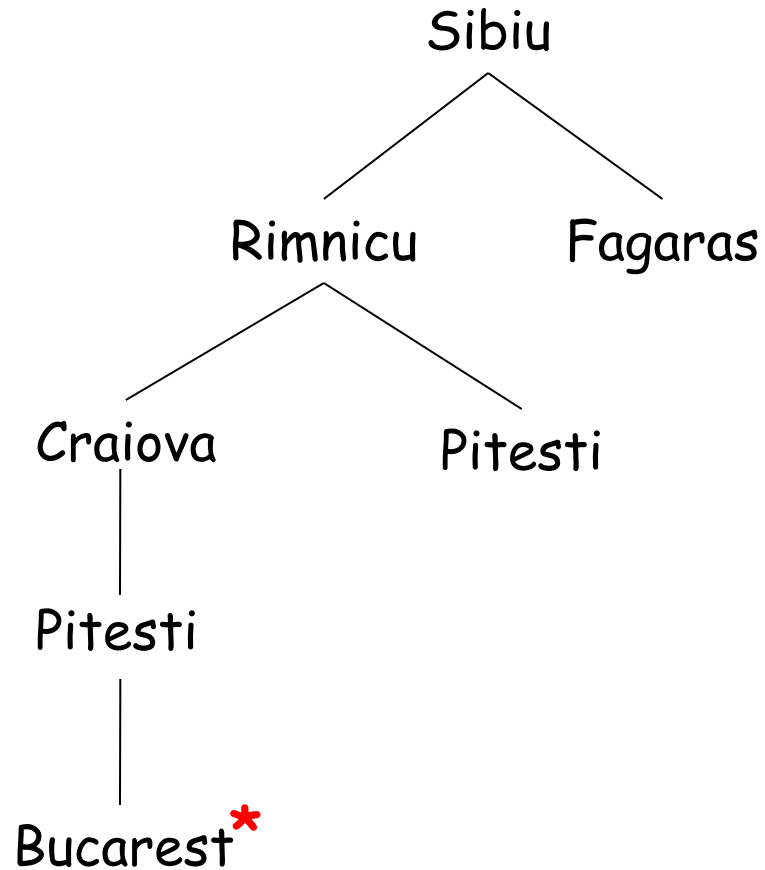
Búsqueda no informada



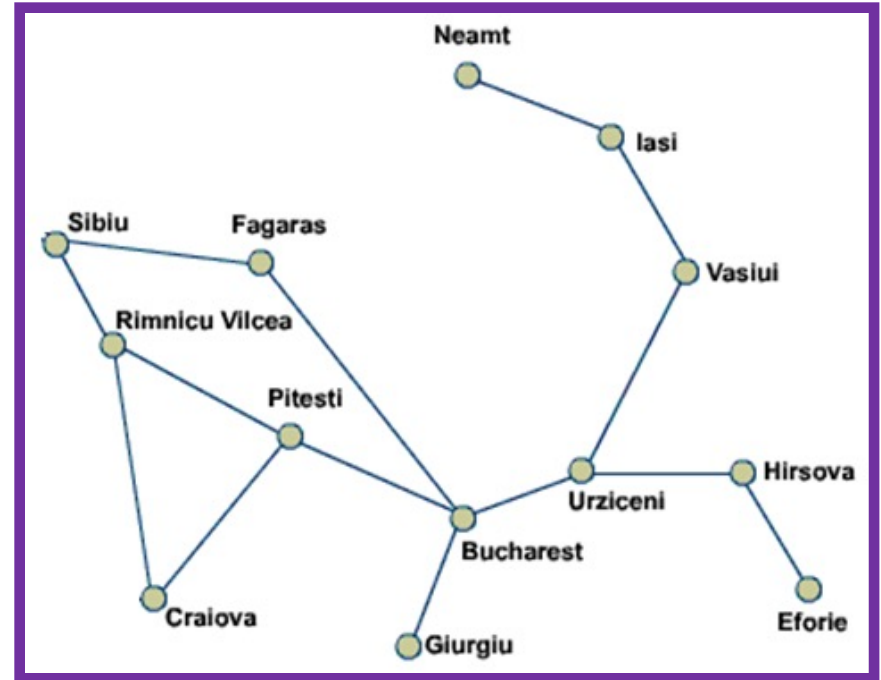
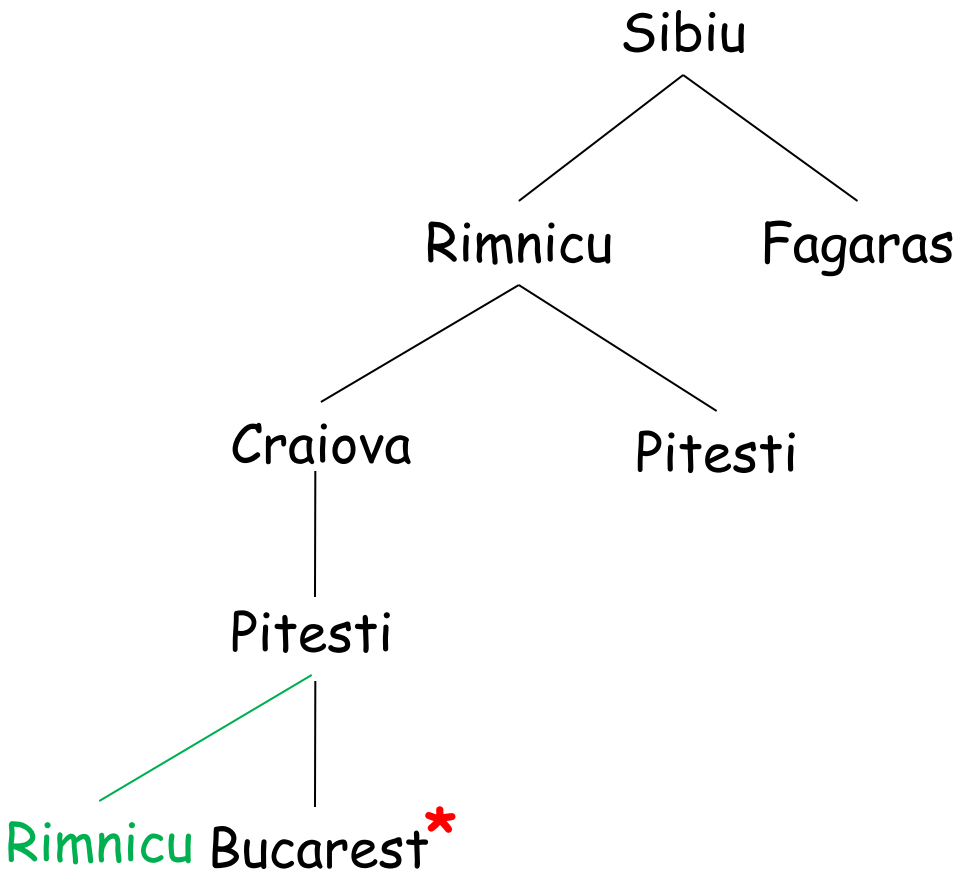
Búsqueda no informada



Búsqueda no informada



Búsqueda no informada



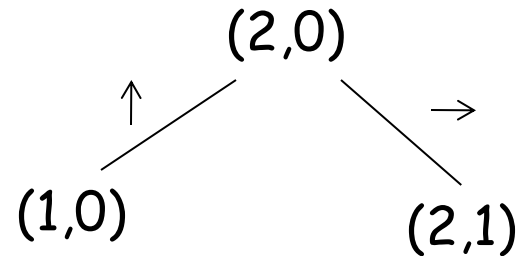
Búsqueda no informada

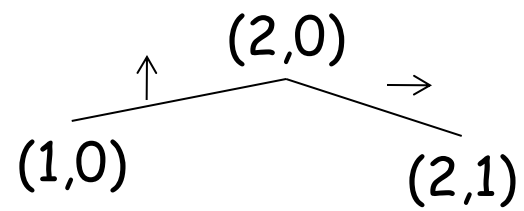
Aplicar búsqueda
preferente por profundidad
evitando ciclos

Aplique los operadores en este
orden:

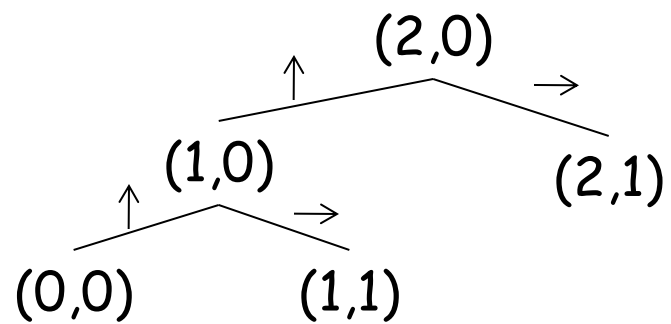
- Arriba
- Abajo
- Izquierda
- Derecha

	0	1	2
0			Fin
1			
2	Inicio		

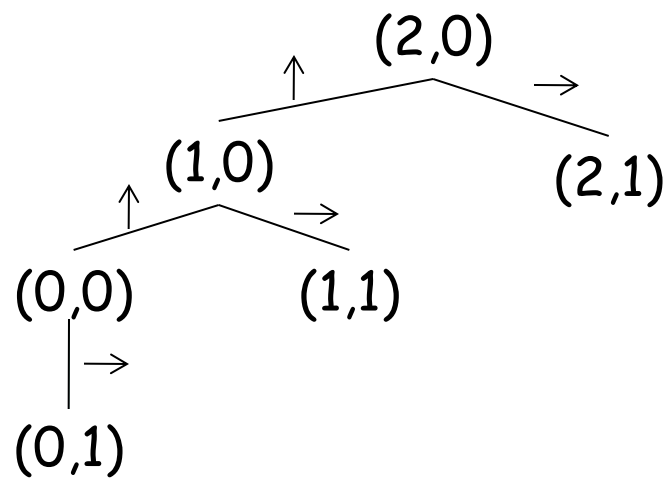




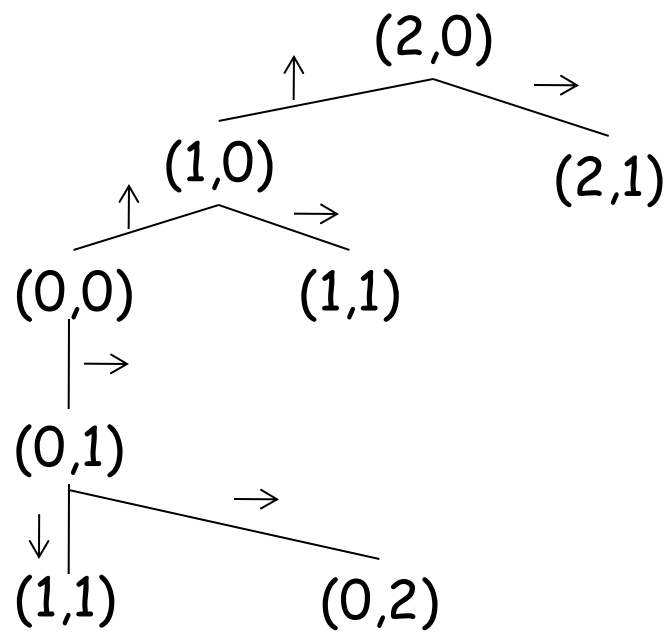
	0	1	2
0			Fin
1			
2	Inicio		



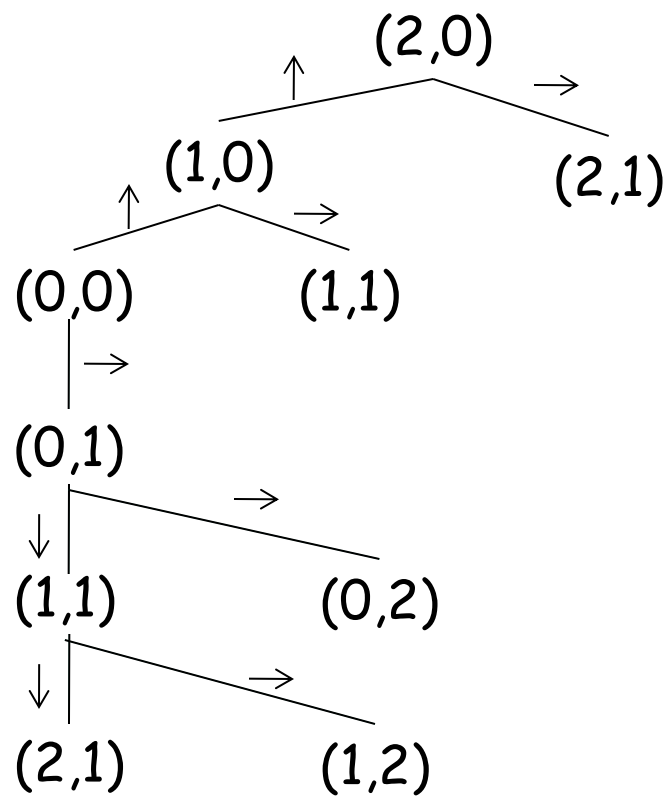
	0	1	2
0			Fin
1			
2	Inicio		



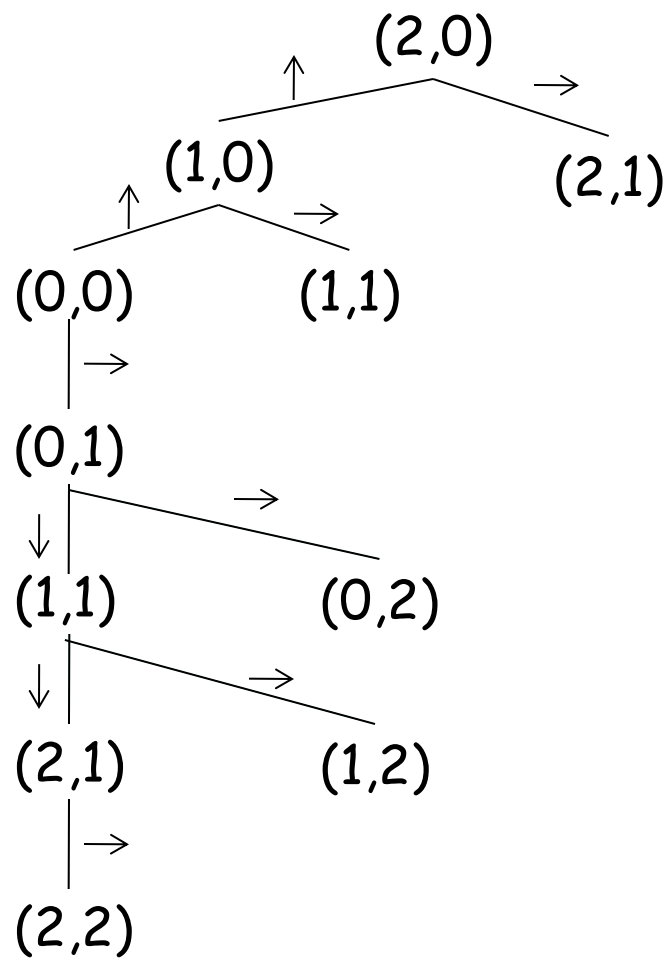
	0	1	2
0			Fin
1			
2	Inicio		



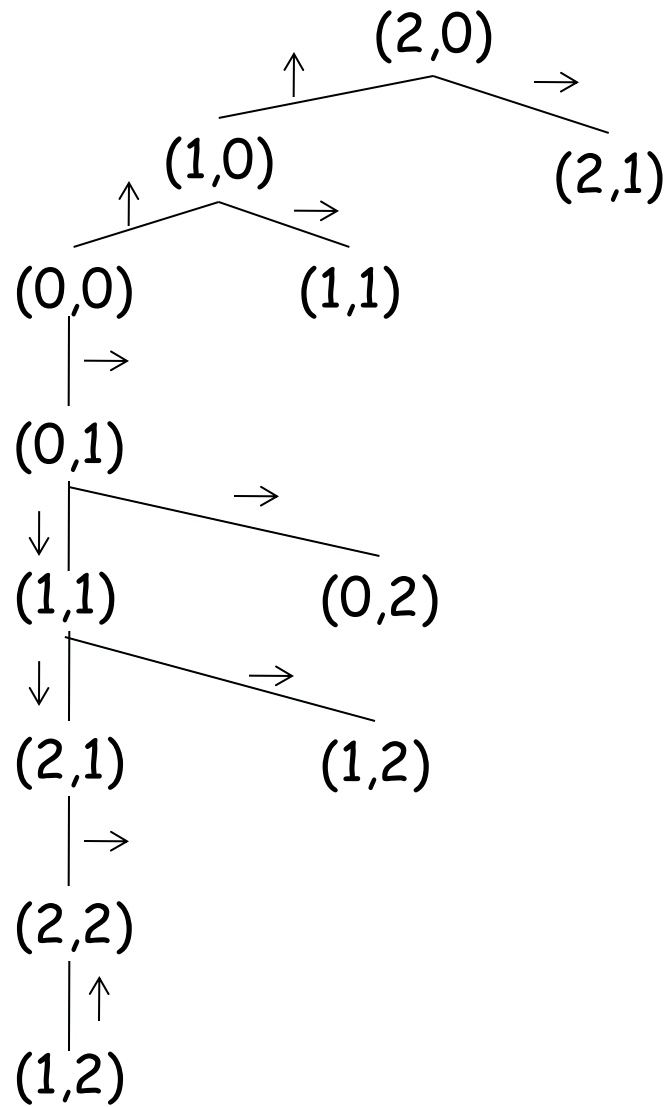
	0	1	2
0			Fin
1			
2	Inicio		



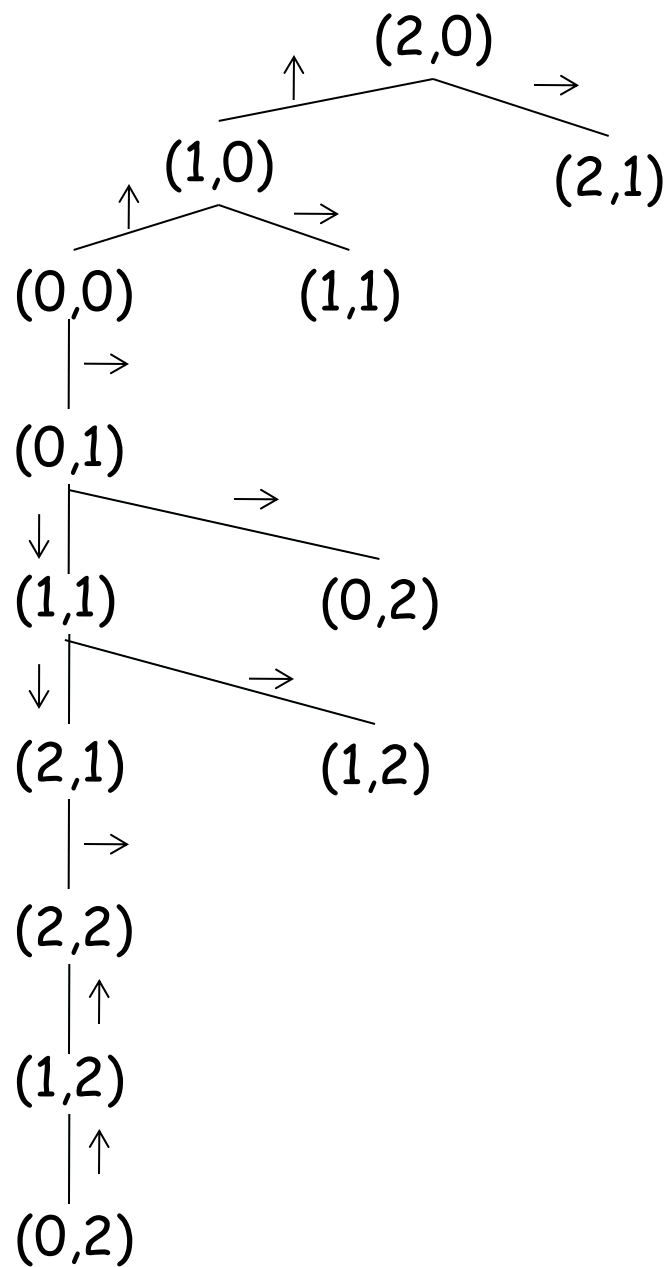
	0	1	2
0			Fin
1			
2	Inicio		



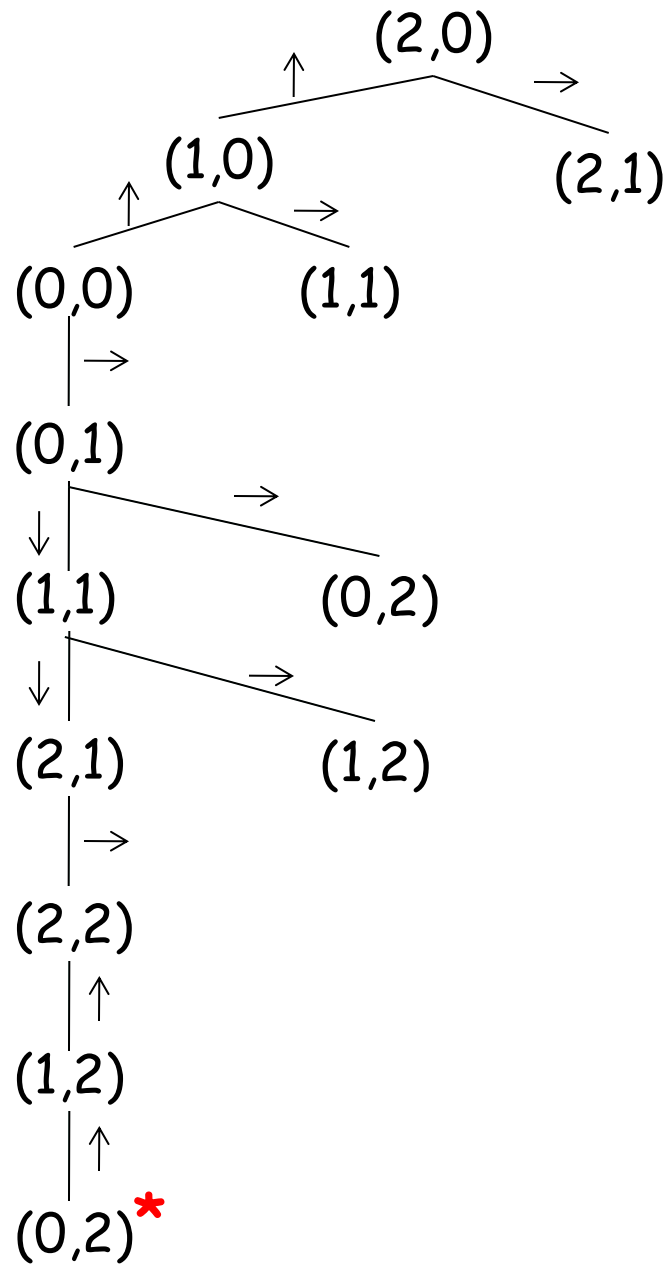
	0	1	2
0			Fin
1			
2	Inicio		



	0	1	2
0			Fin
1			
2	Inicio		



	0	1	2
0			Fin
1			
2	Inicio		

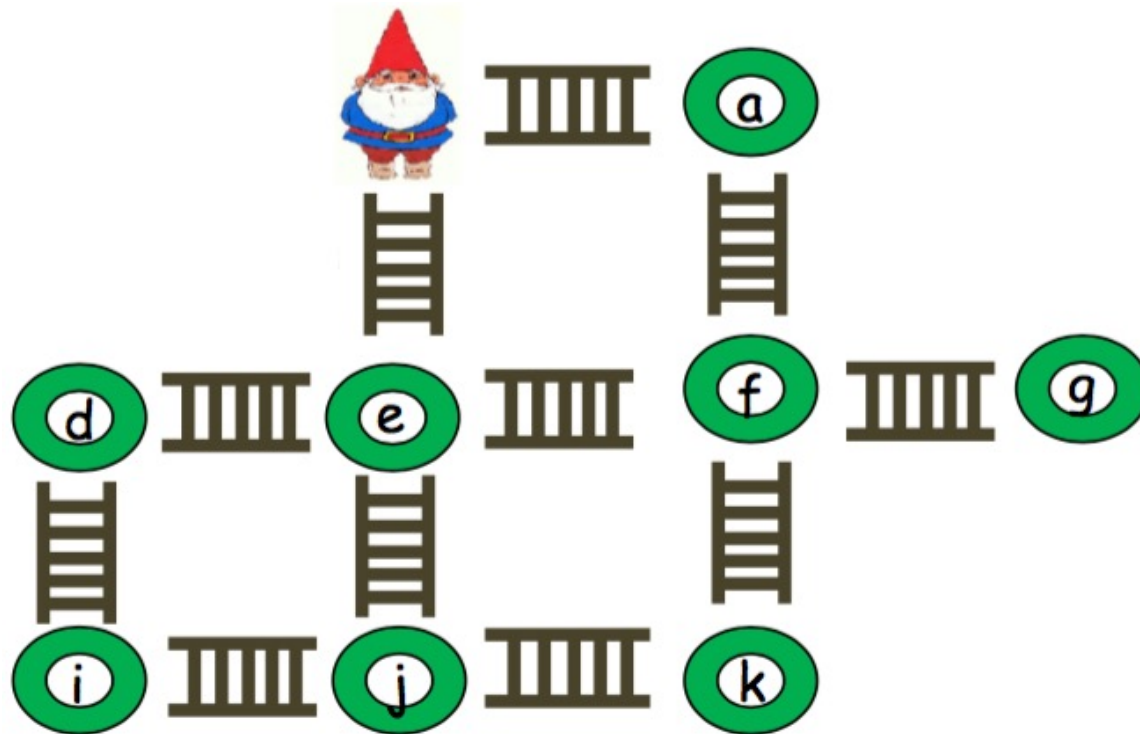


	0	1	2
0			Fin
1			
2	Inicio		

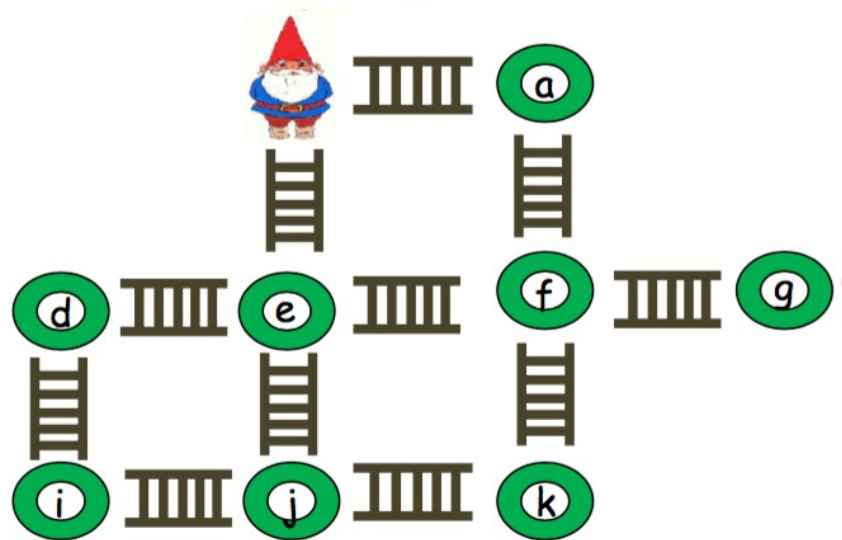
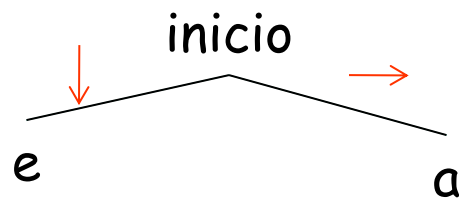


0
1
2

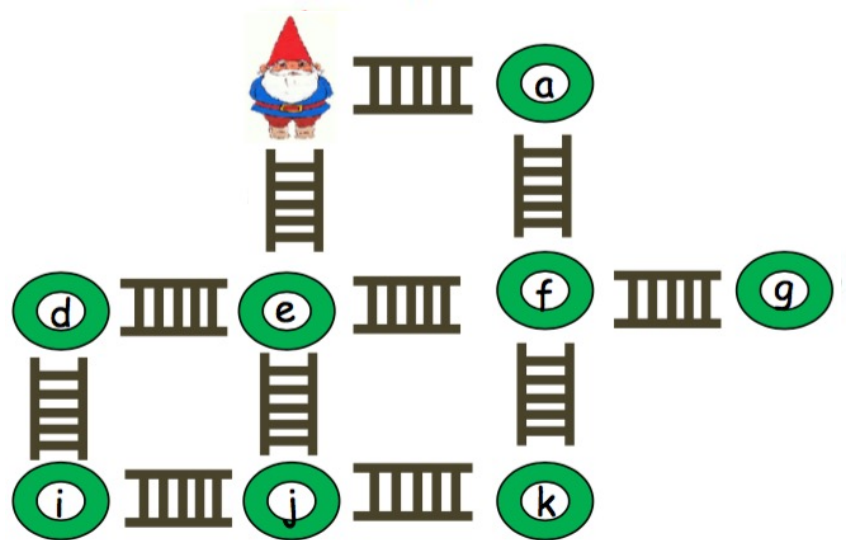
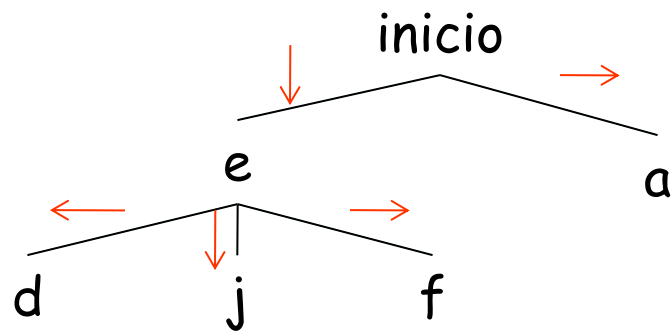
El nomo inteligente. El problema consiste en encontrar un camino que le permita al nomo llegar al sitio g. Aplicar búsqueda por profundidad evitando ciclos



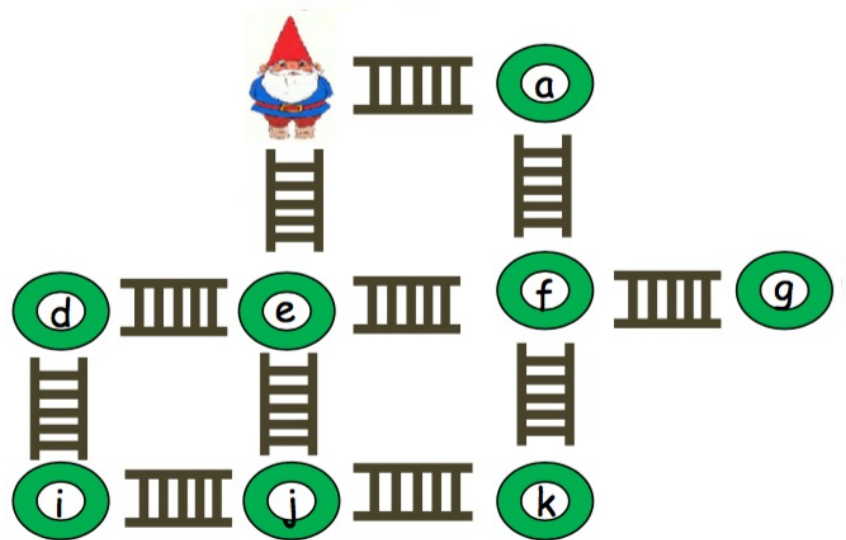
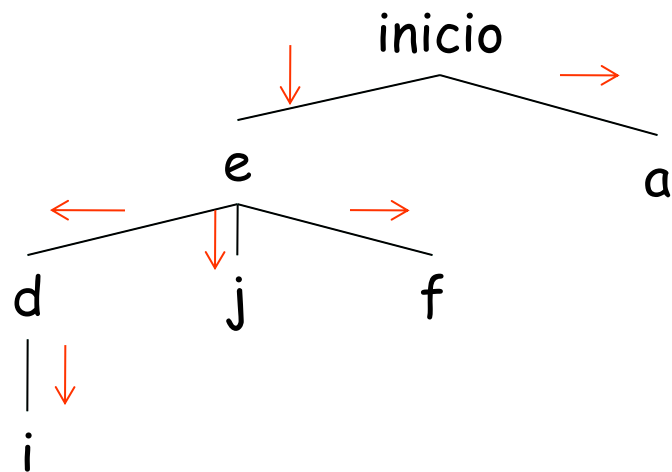
Aplique los operadores en el orden: \leftarrow , \downarrow , \rightarrow , \uparrow



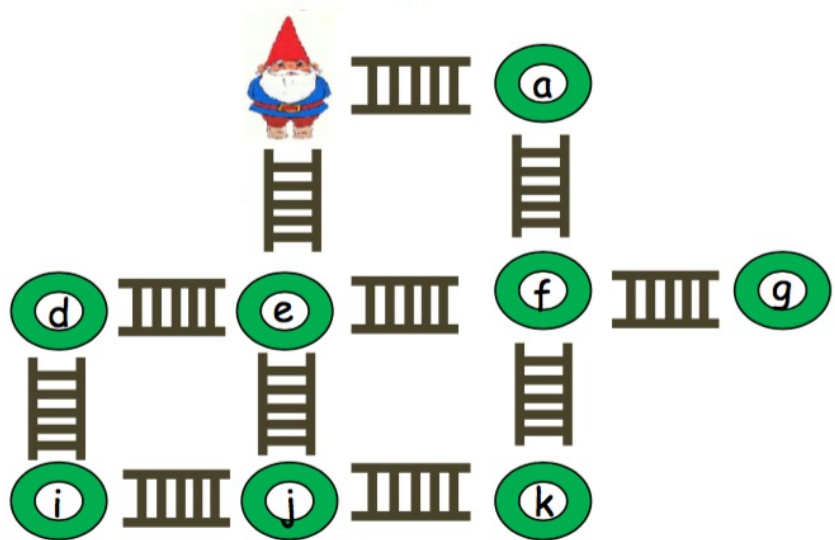
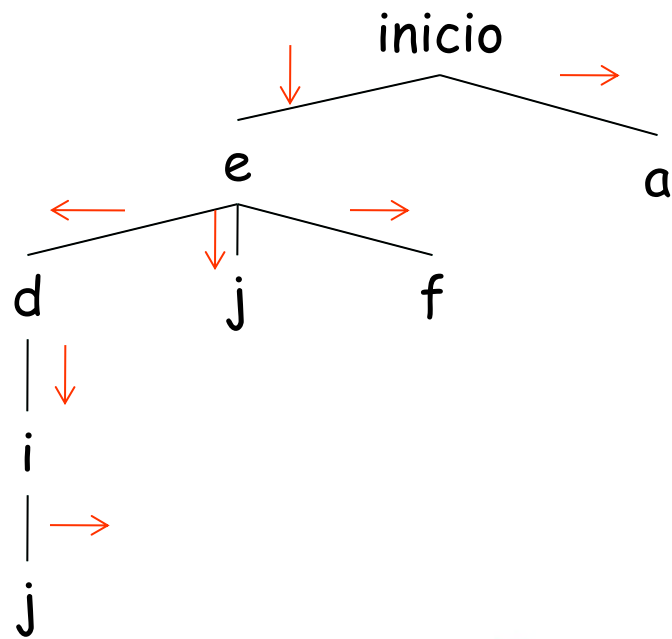
←, ↓, →, ↑



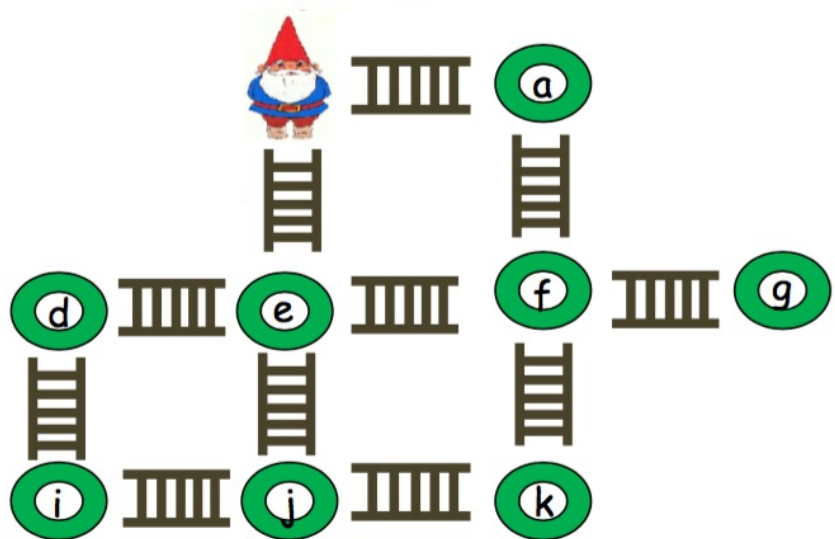
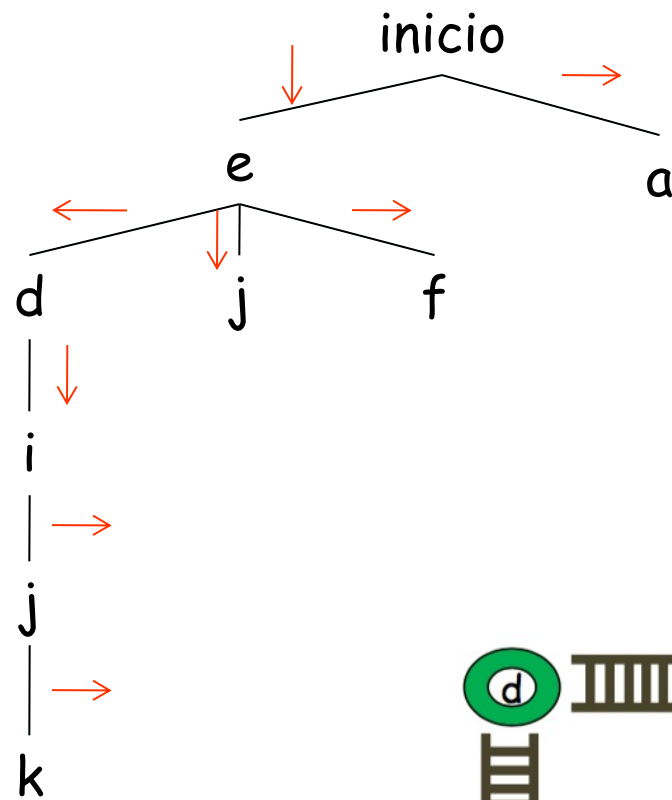
←, ↓, →, ↑



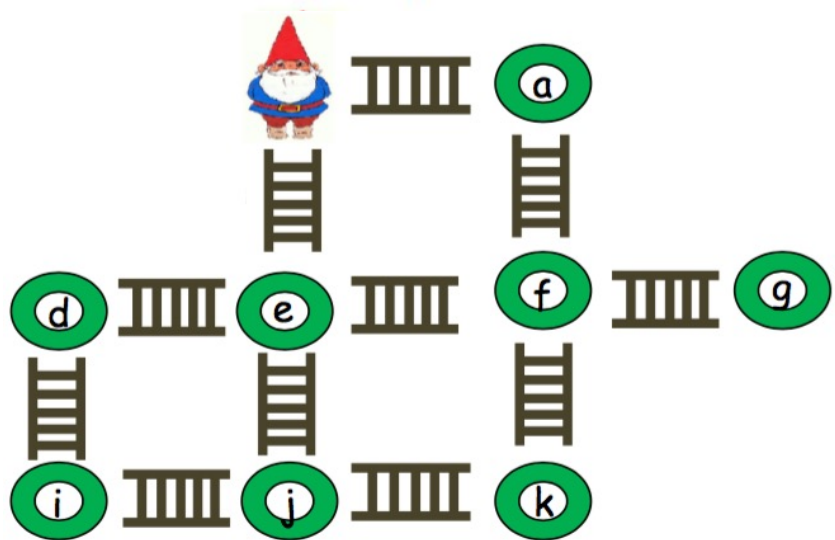
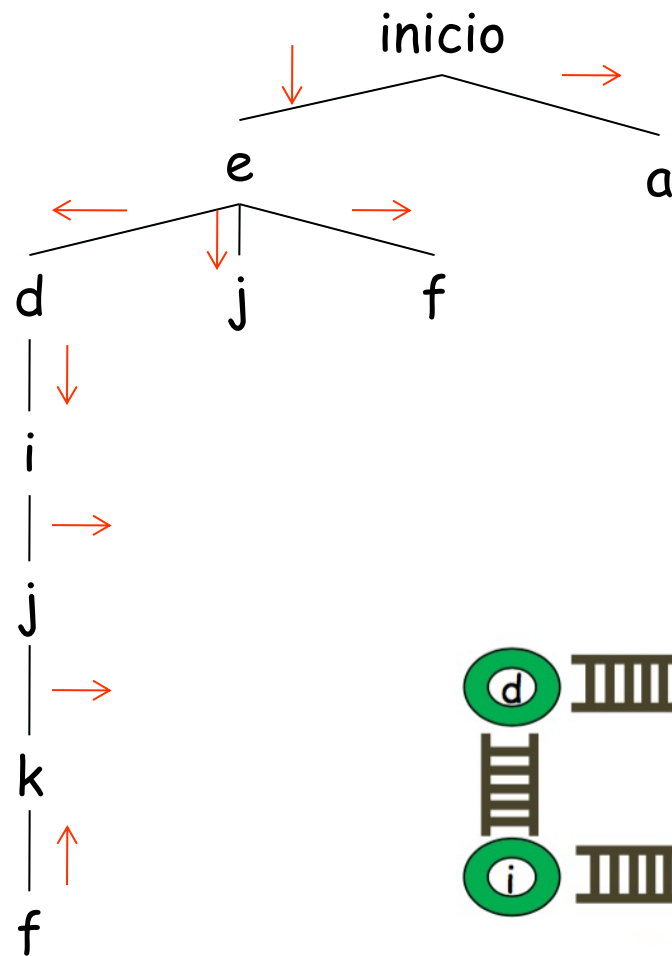
←, ↓, →, ↑



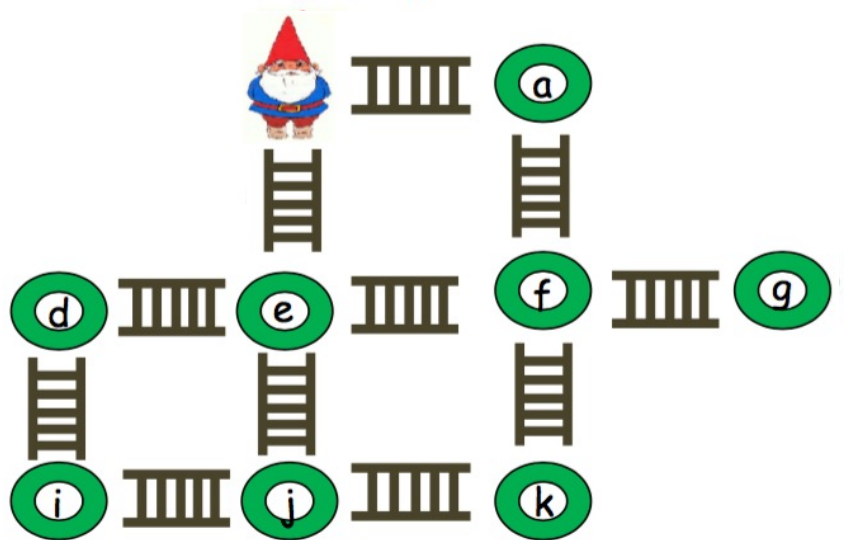
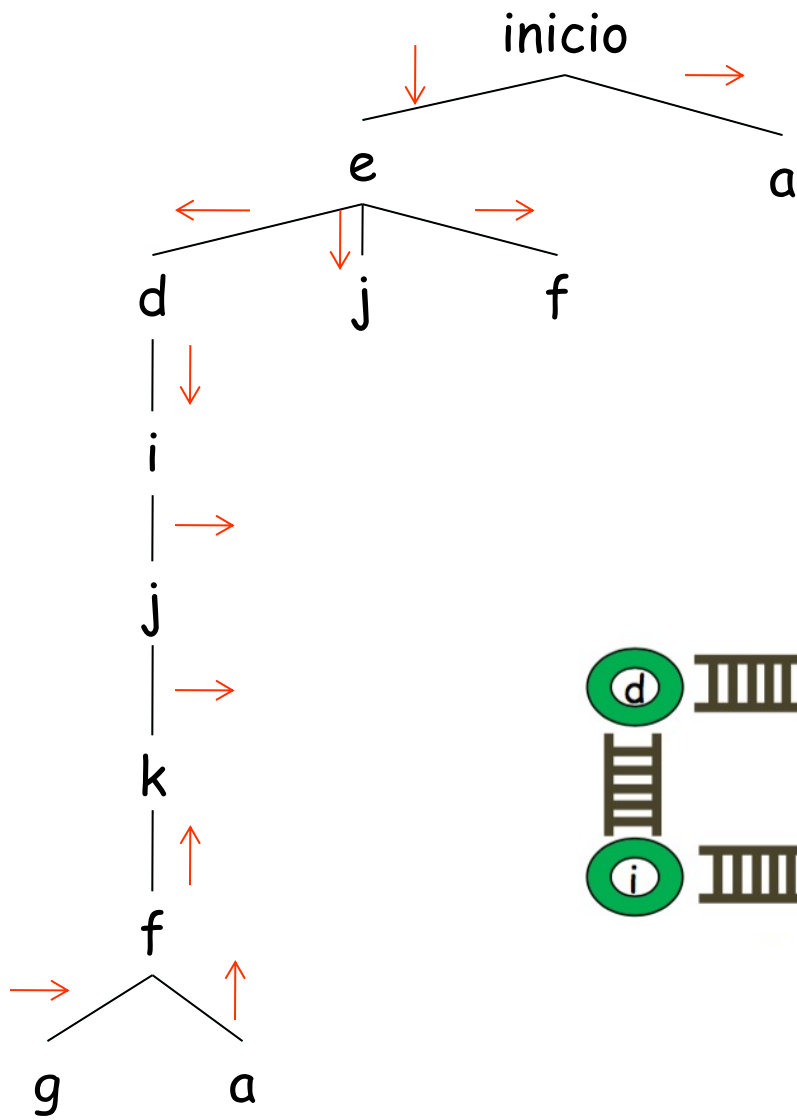
←, ↓, →, ↑



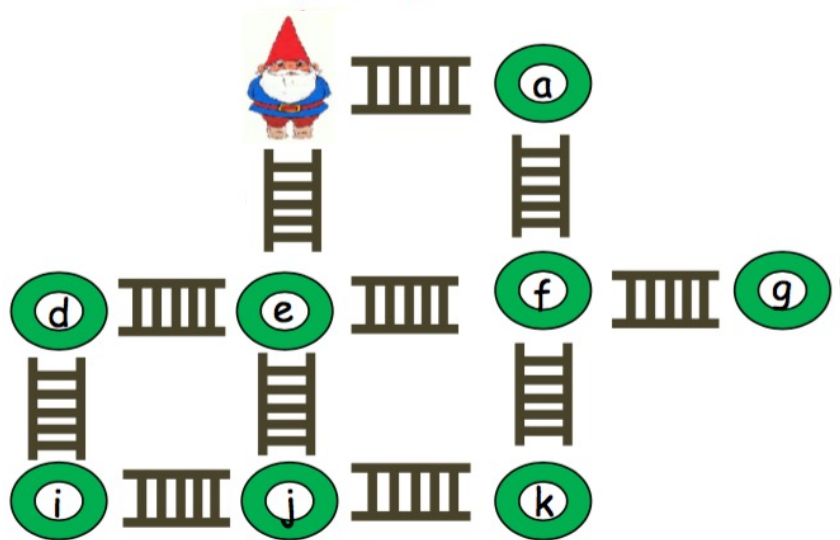
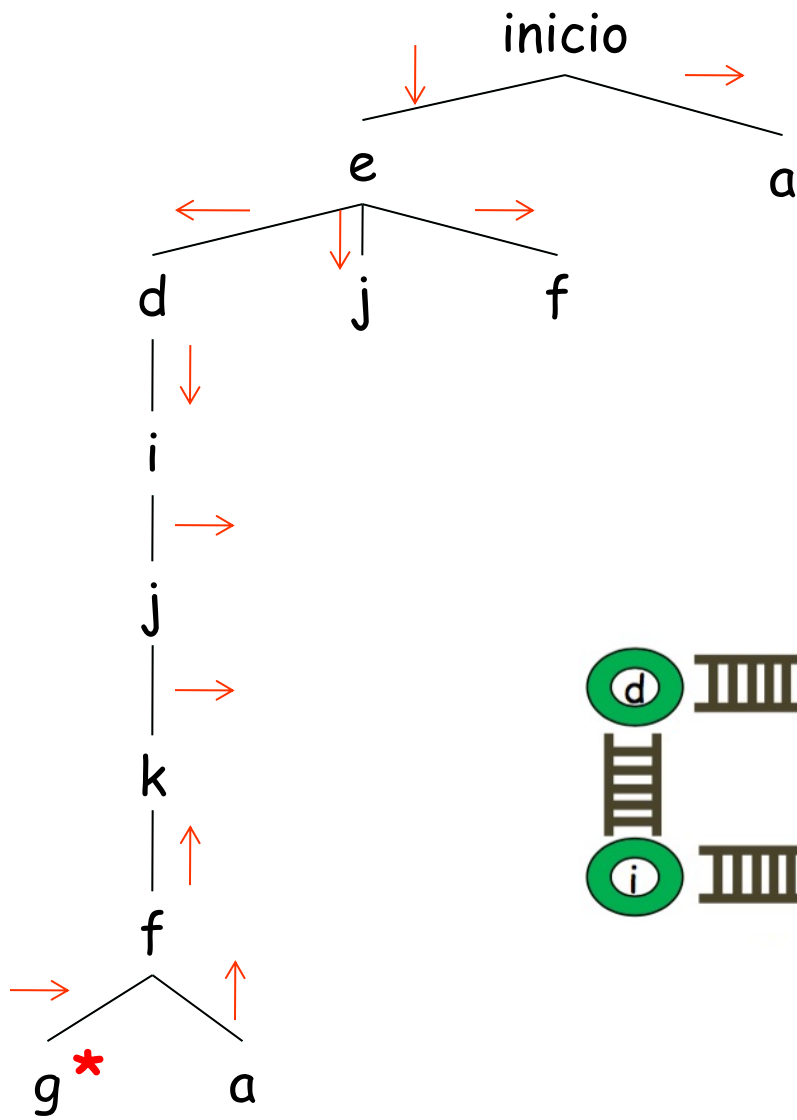
←, ↓, →, ↑



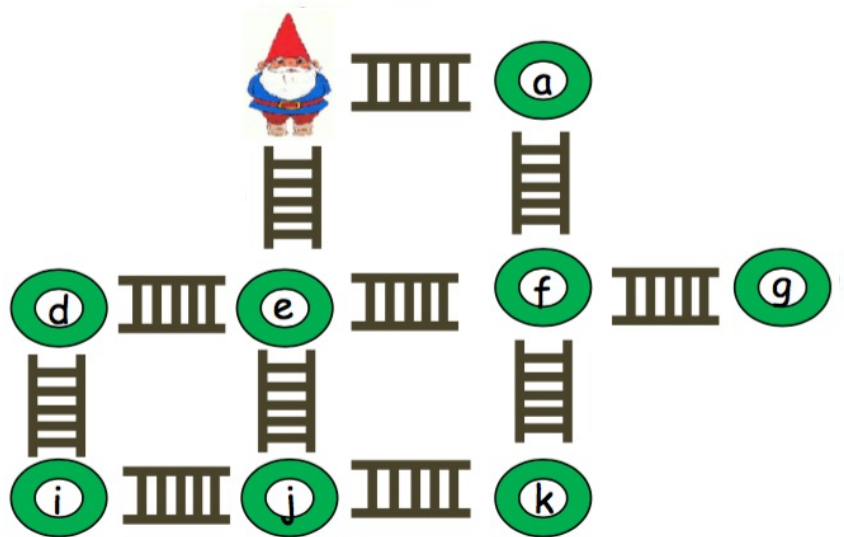
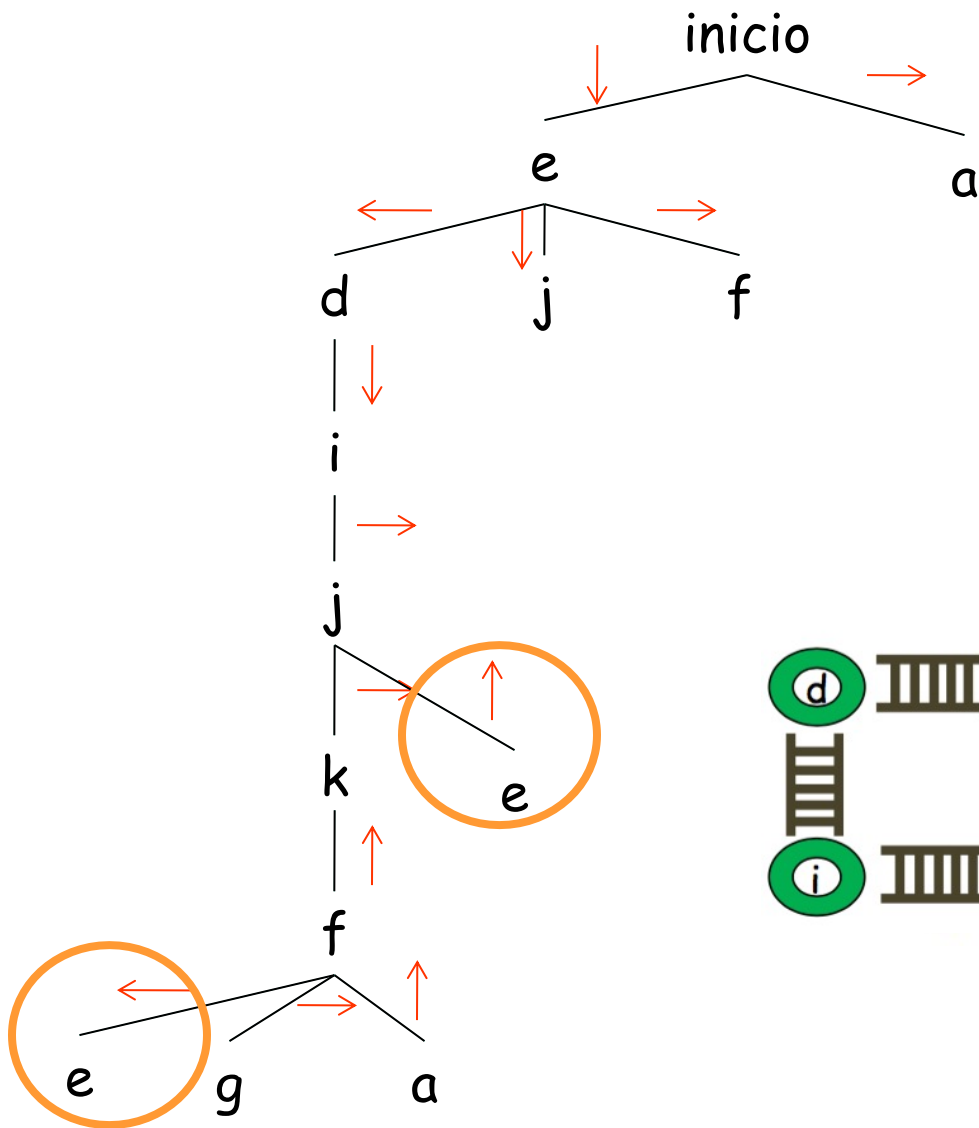
←, ↓, →, ↑



←, ↓, →, ↑

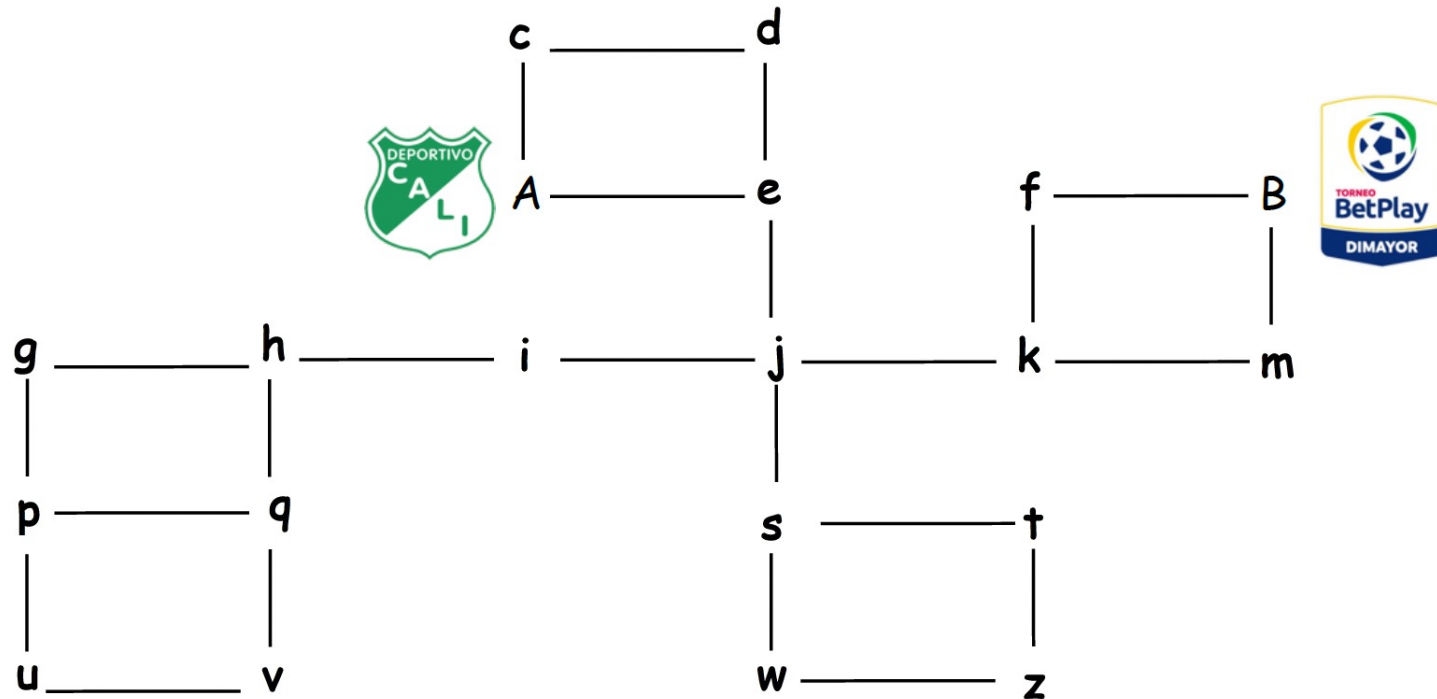


←, ↓, →, ↑

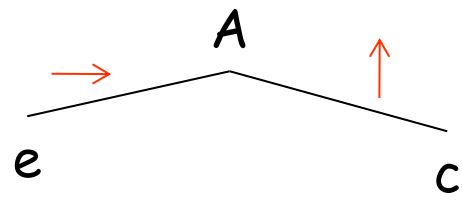


←, ↓, →, ↑

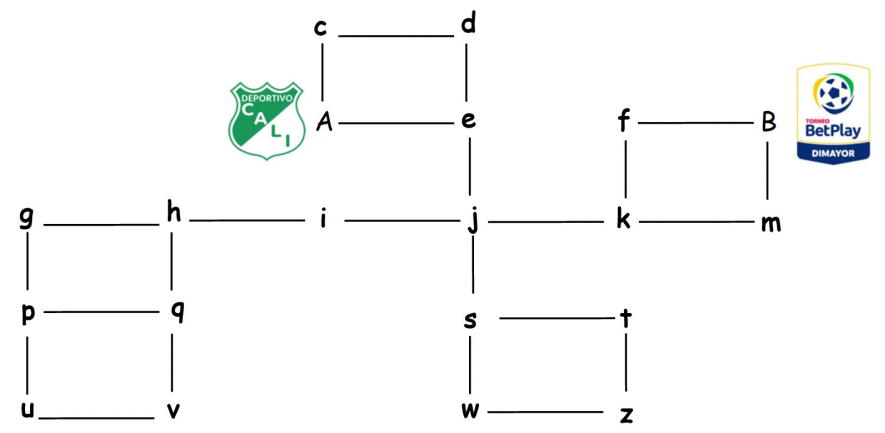
2. [I.L 1.3 (7pts)] El deportivo Cali está buscando la B y para ello usará el algoritmo por profundidad evitando ciclos. Considere el ambiente que se muestra en la figura donde el punto inicial es A y se tiene como objetivo llegar al sitio marcado con la letra B. Debe aplicar los operadores en el orden $\leftarrow, \downarrow, \rightarrow, \uparrow$.

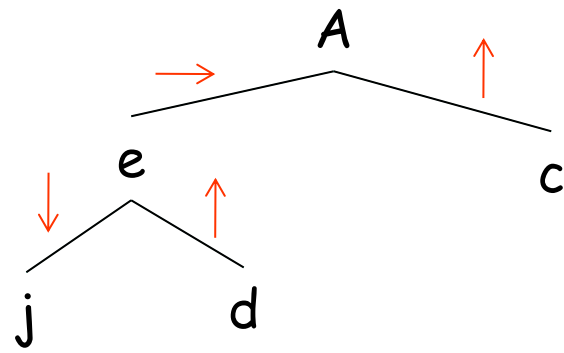


Muestre el árbol construido y la solución encontrada (el camino).

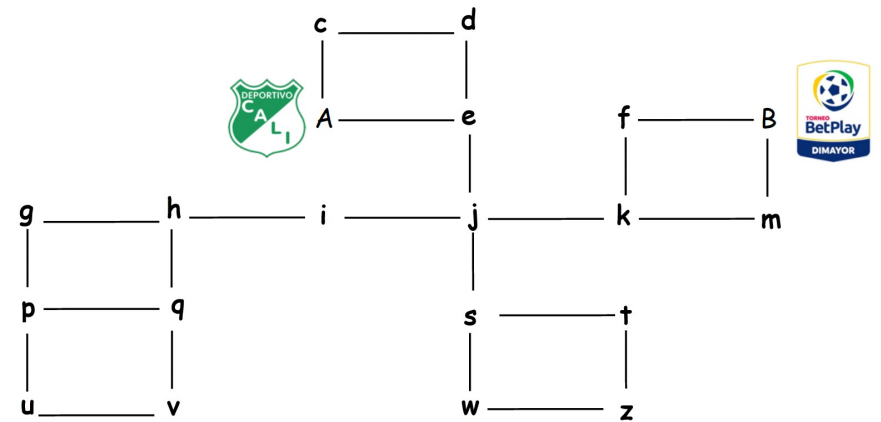


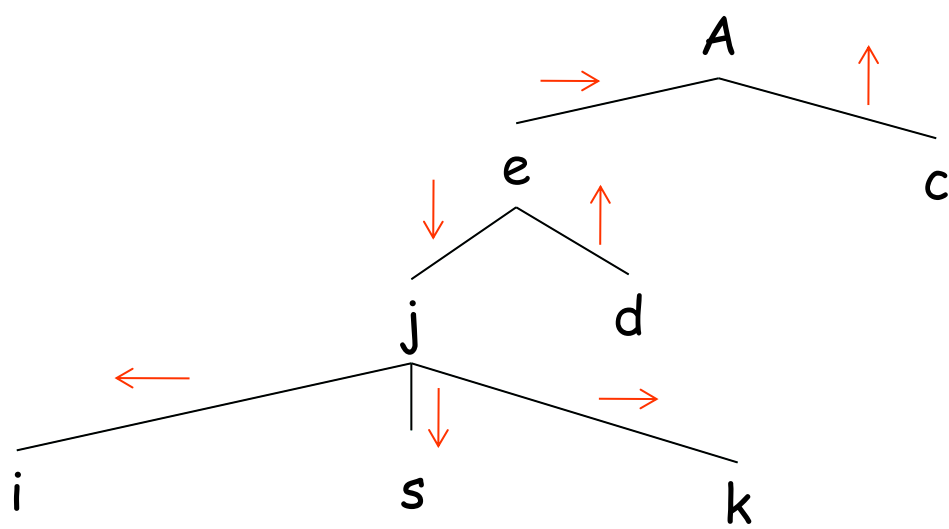
←, ↓, →, ↑



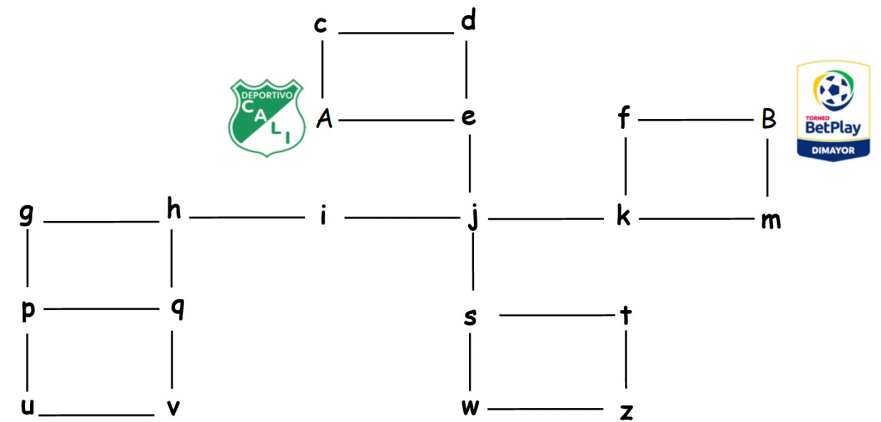


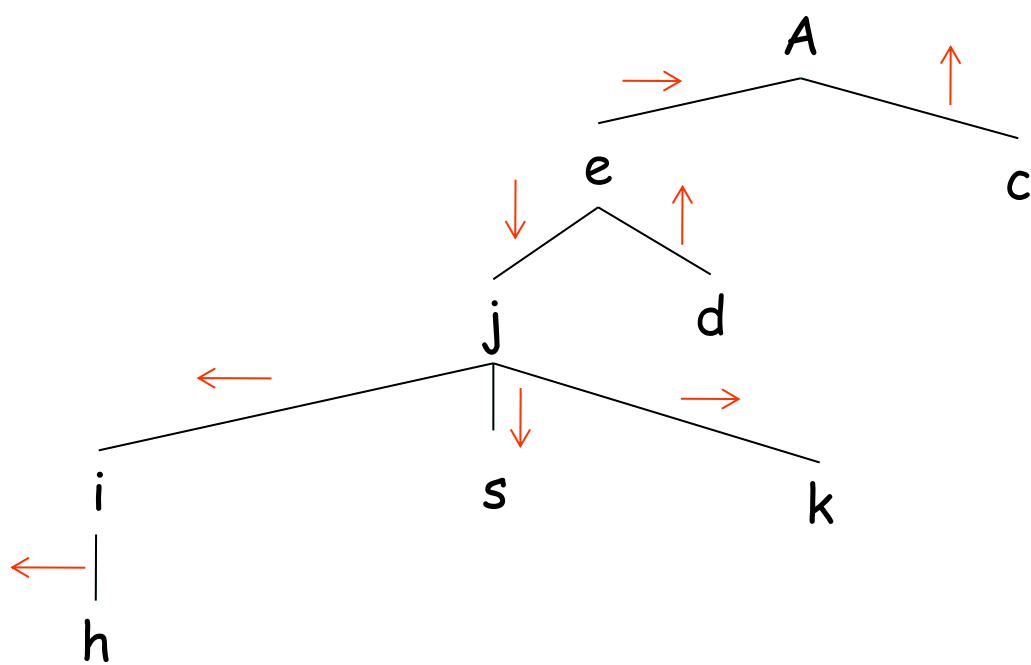
←, ↓, →, ↑



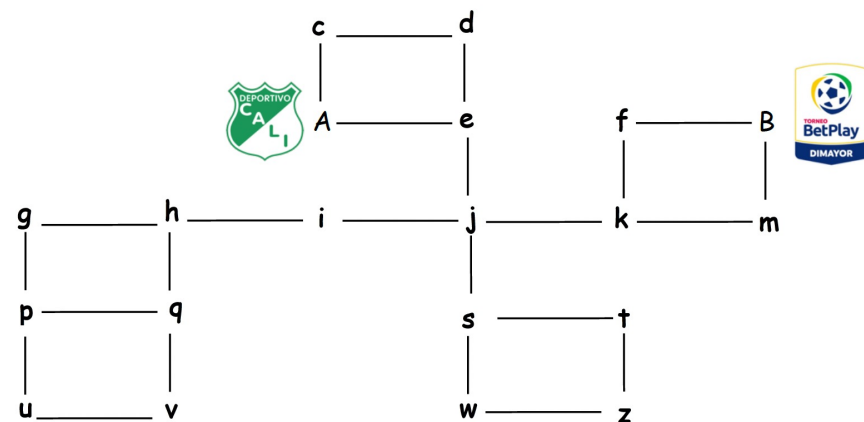


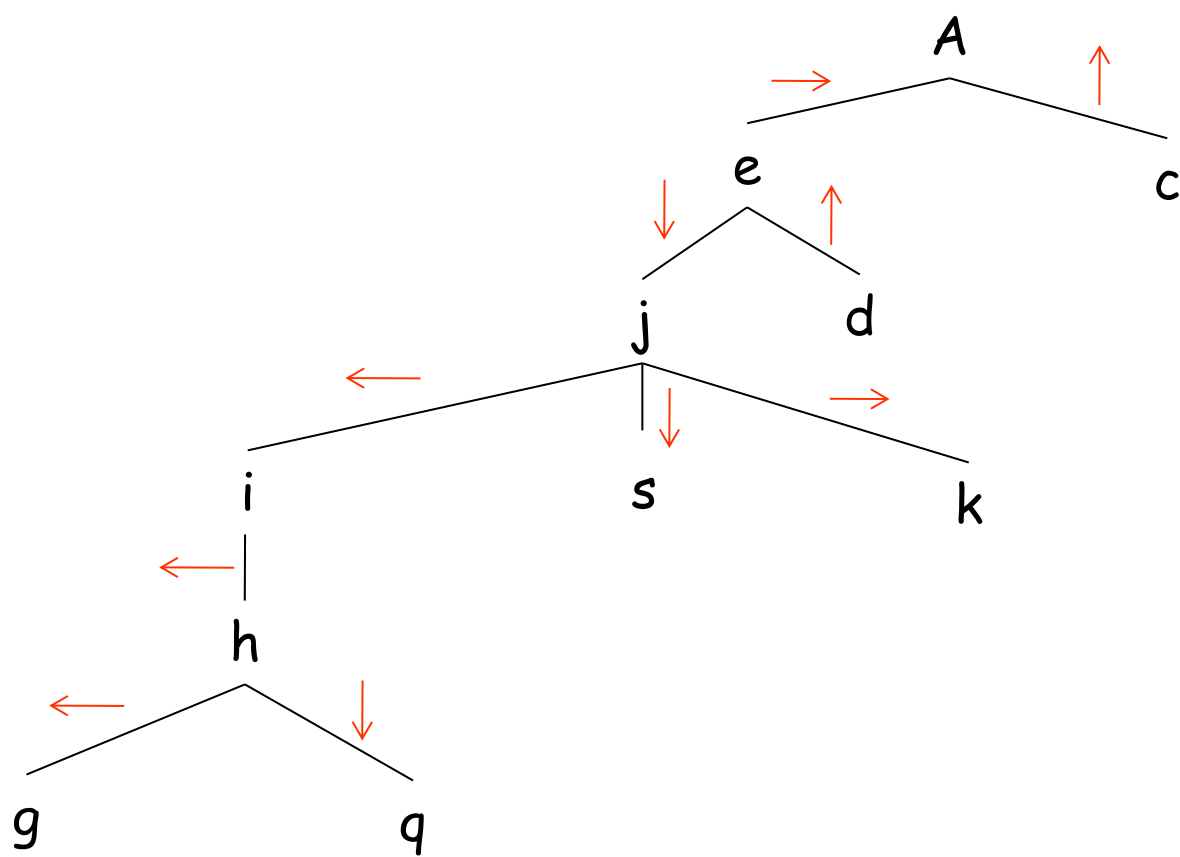
←, ↓, →, ↑



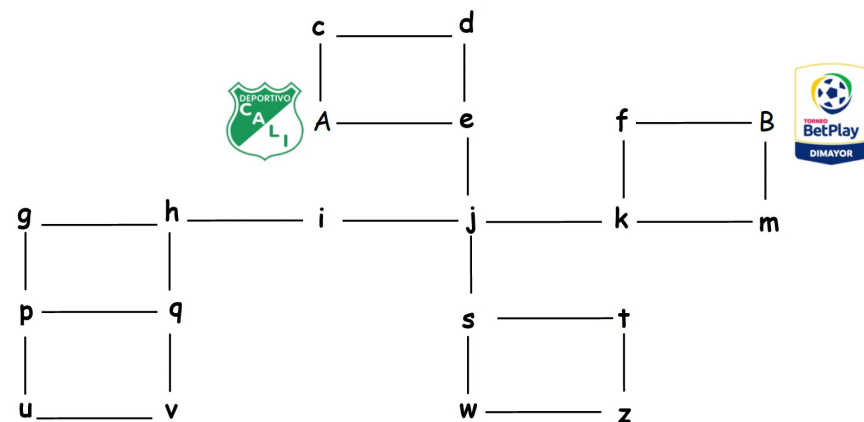


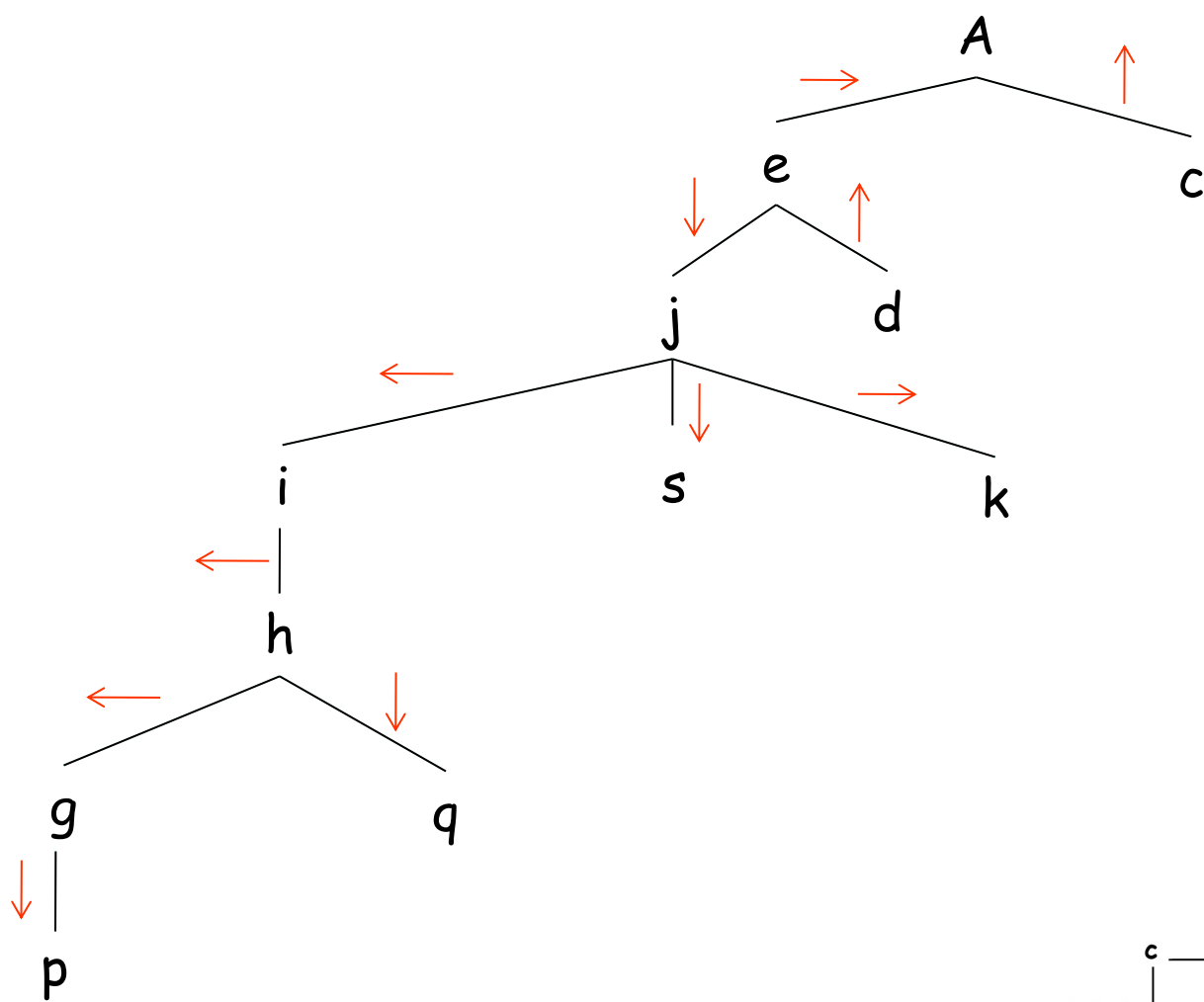
←, ↓, →, ↑



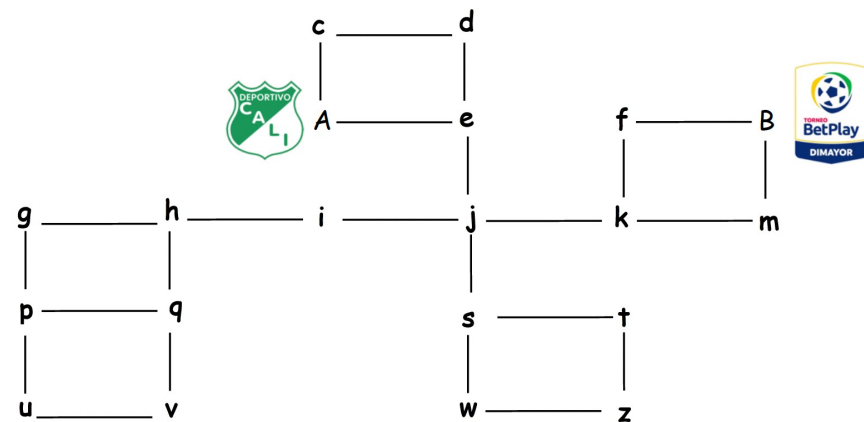


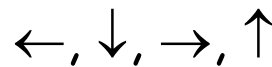
←, ↓, →, ↑

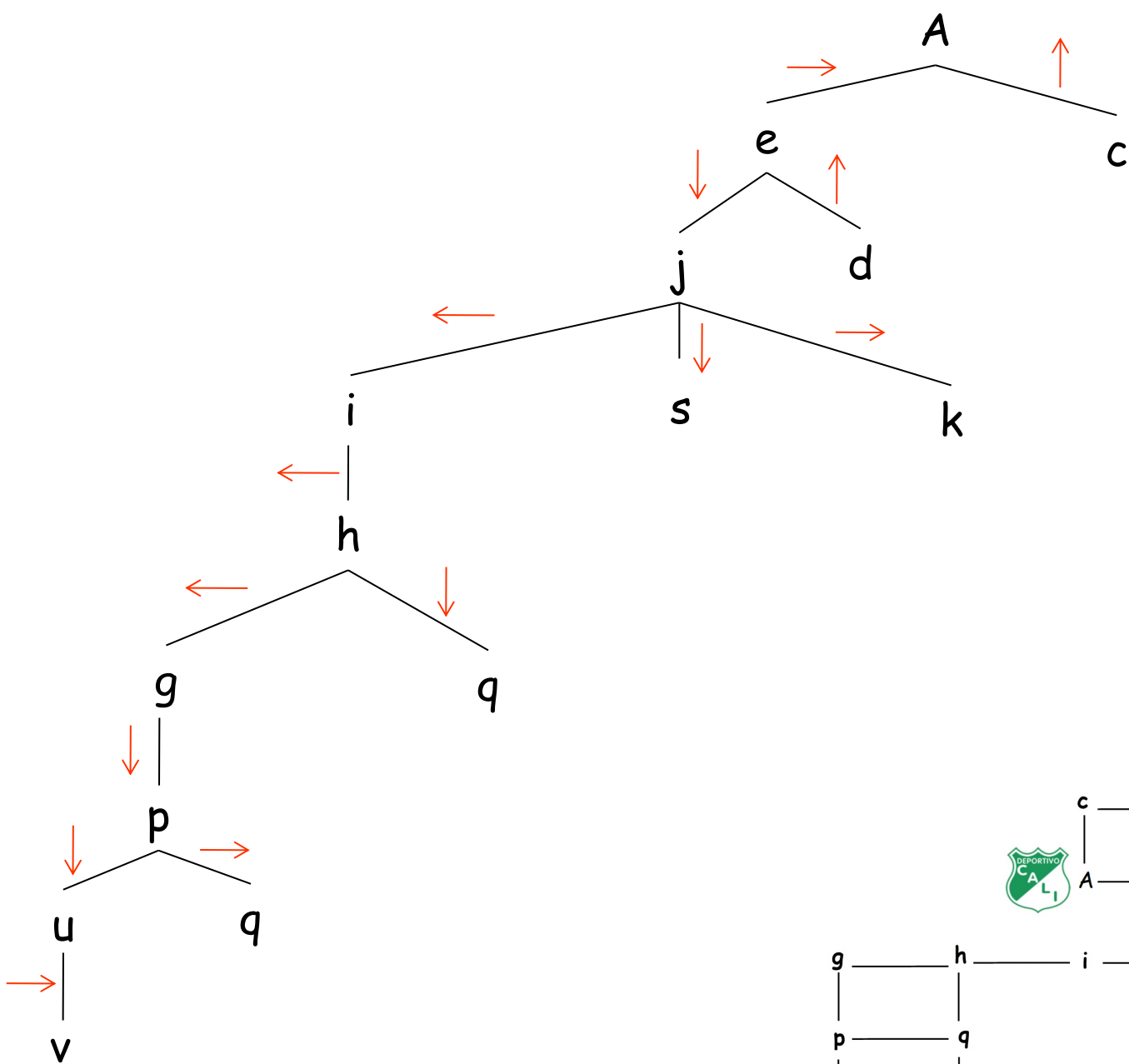




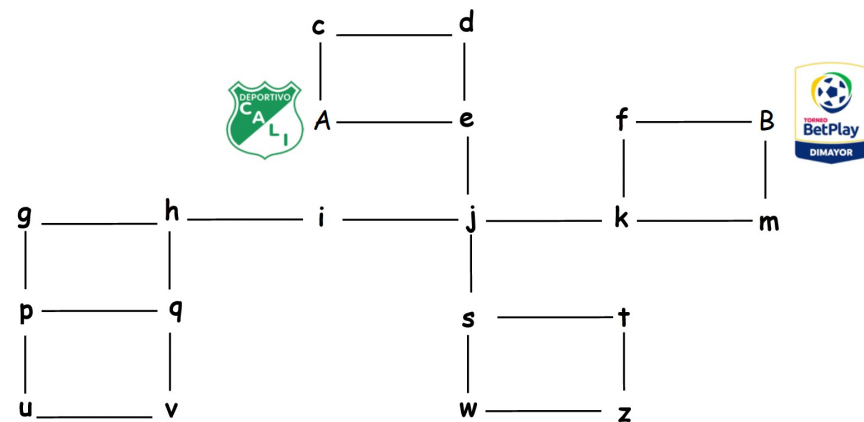
←, ↓, →, ↑

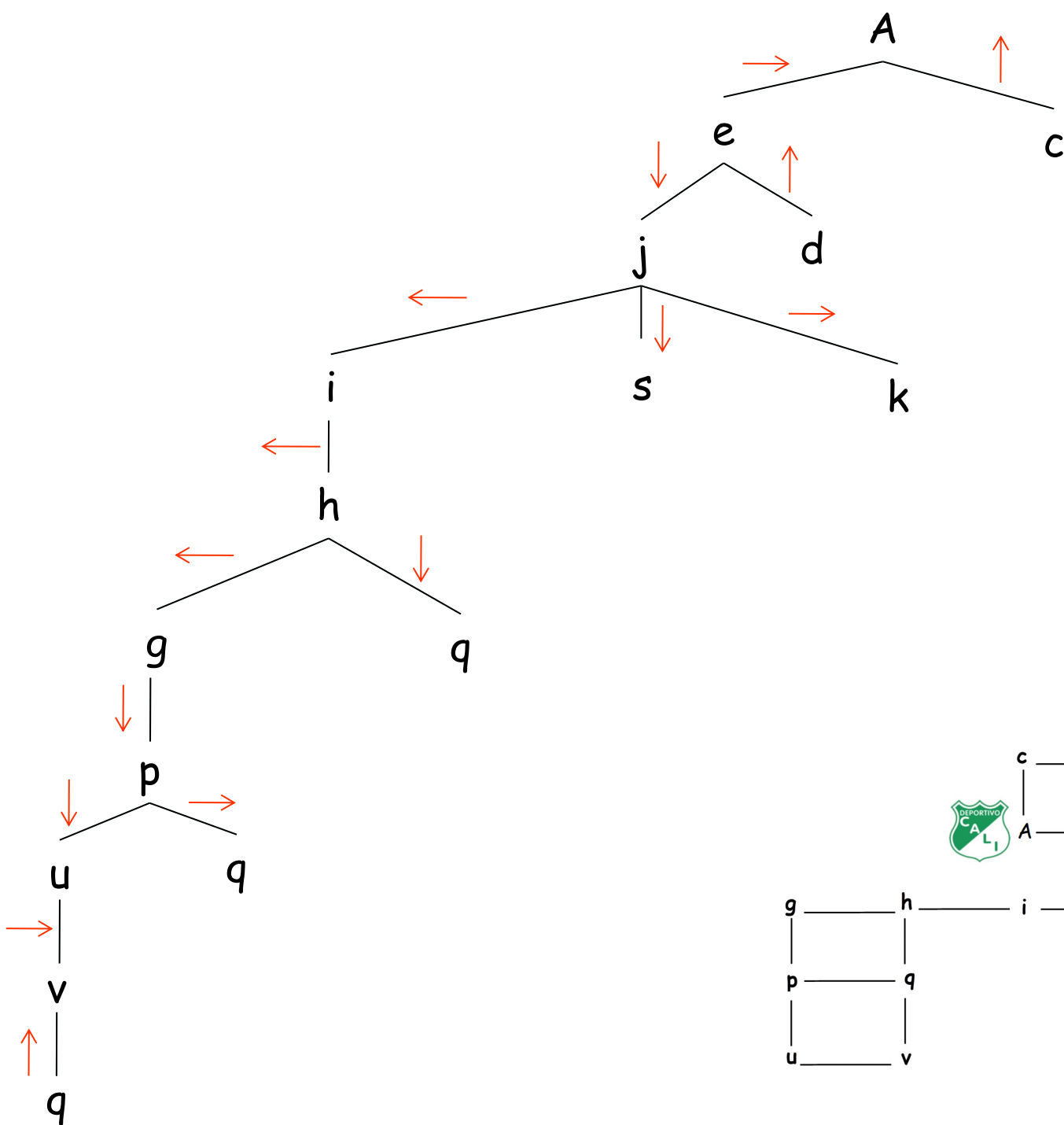




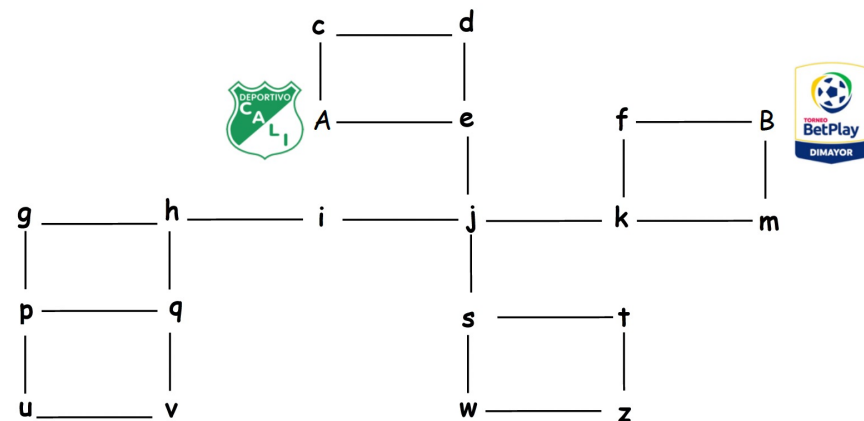


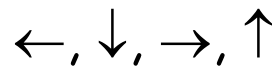
←, ↓, →, ↑

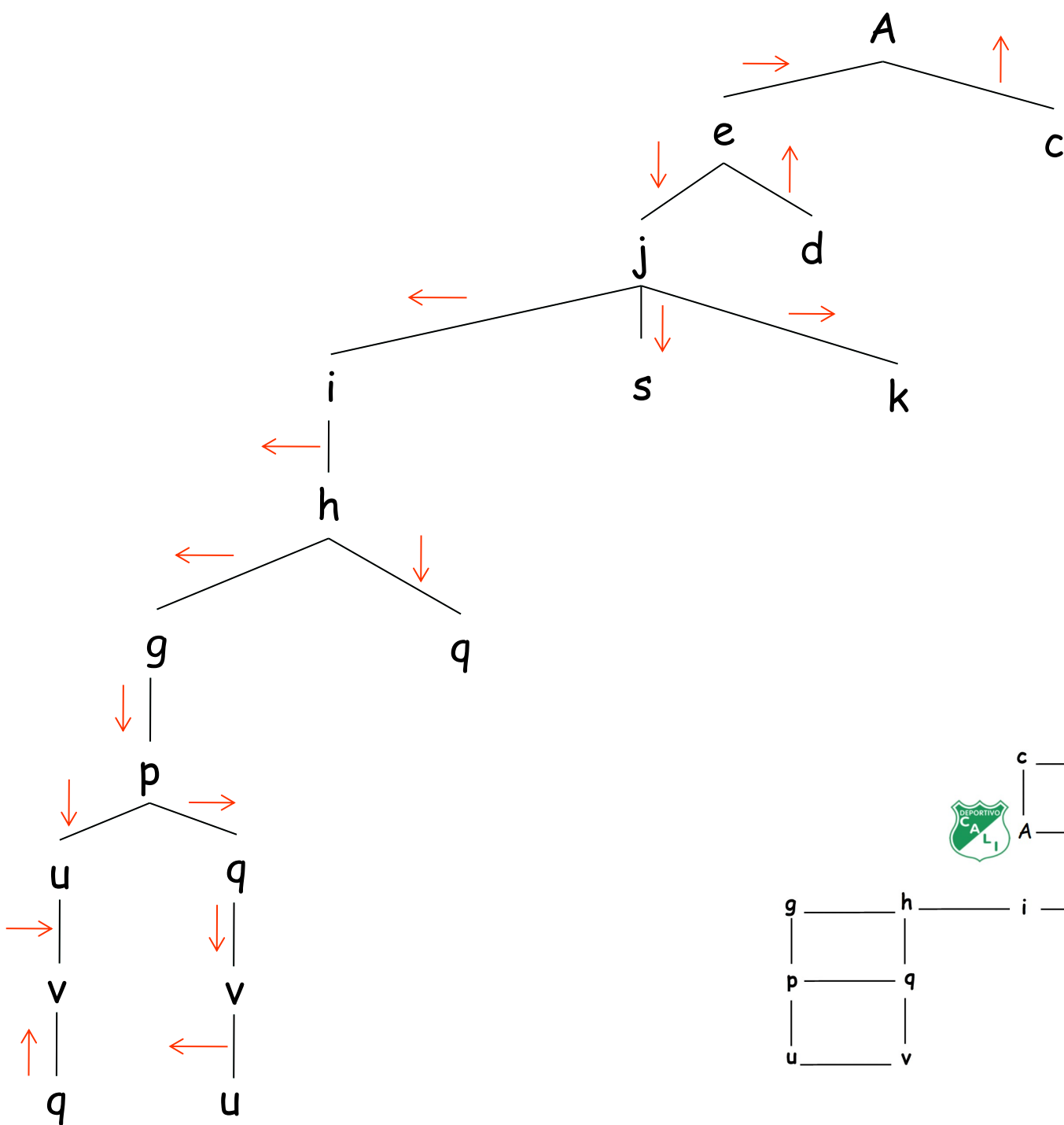




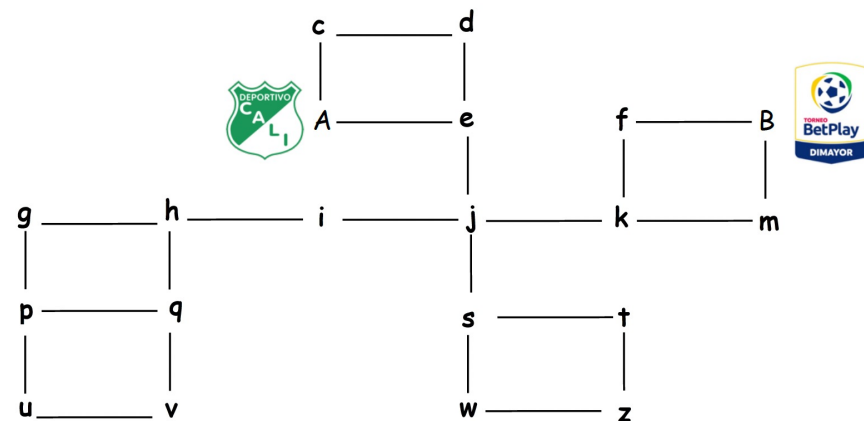
←, ↓, →, ↑

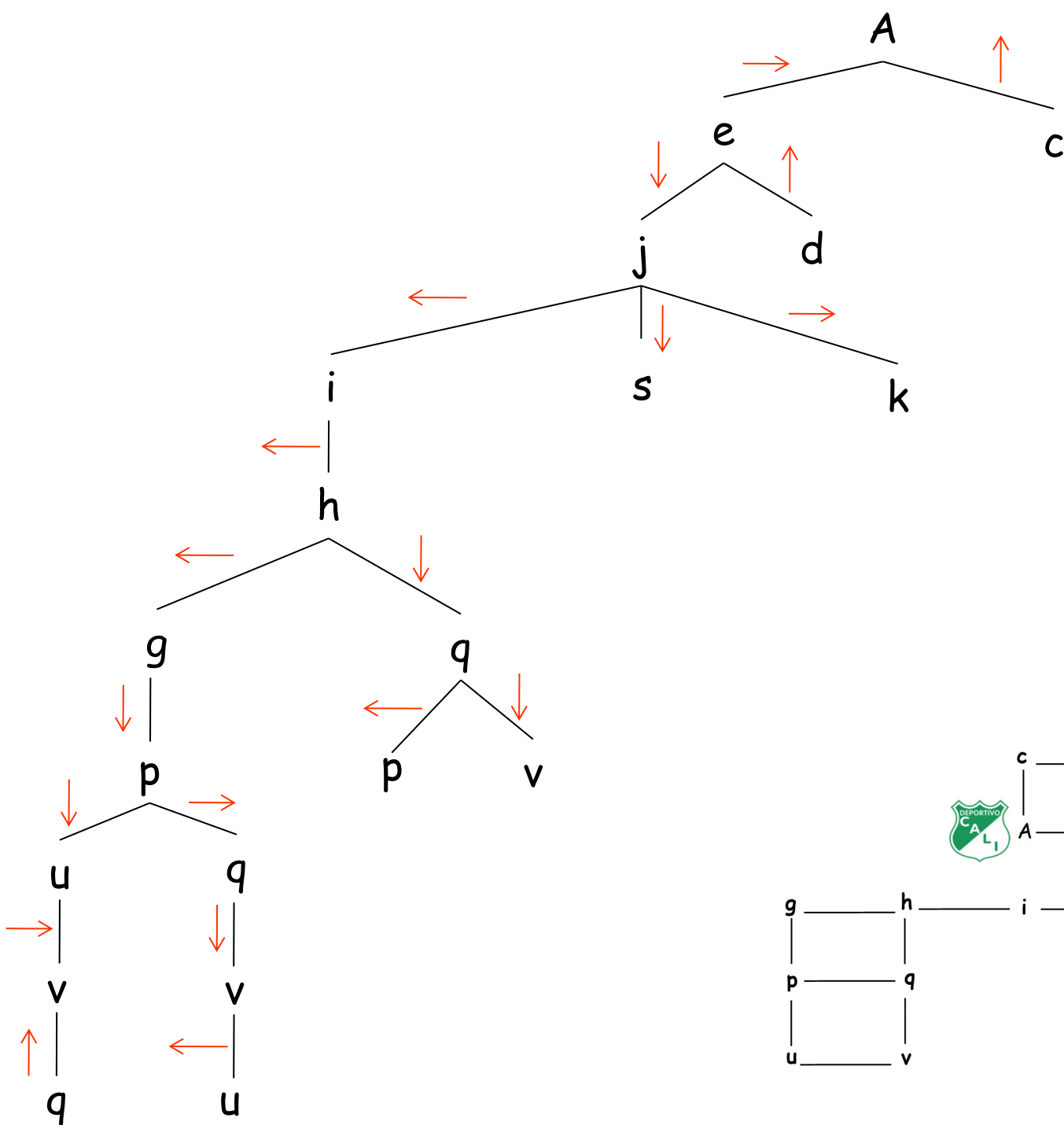




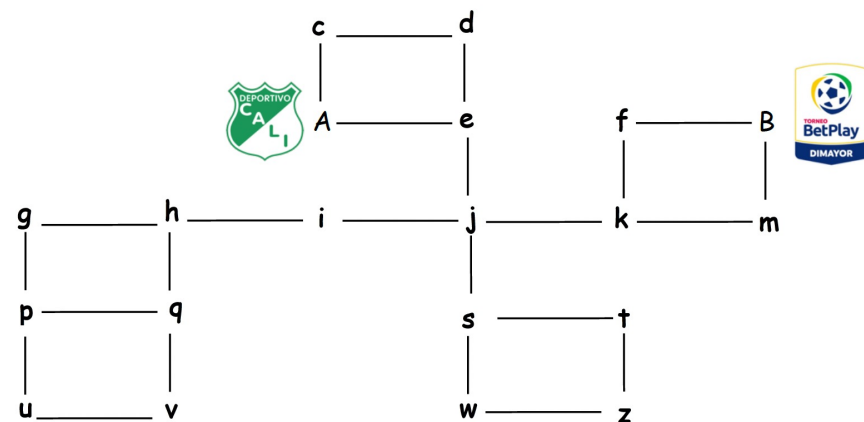


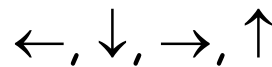
←, ↓, →, ↑

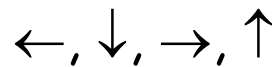


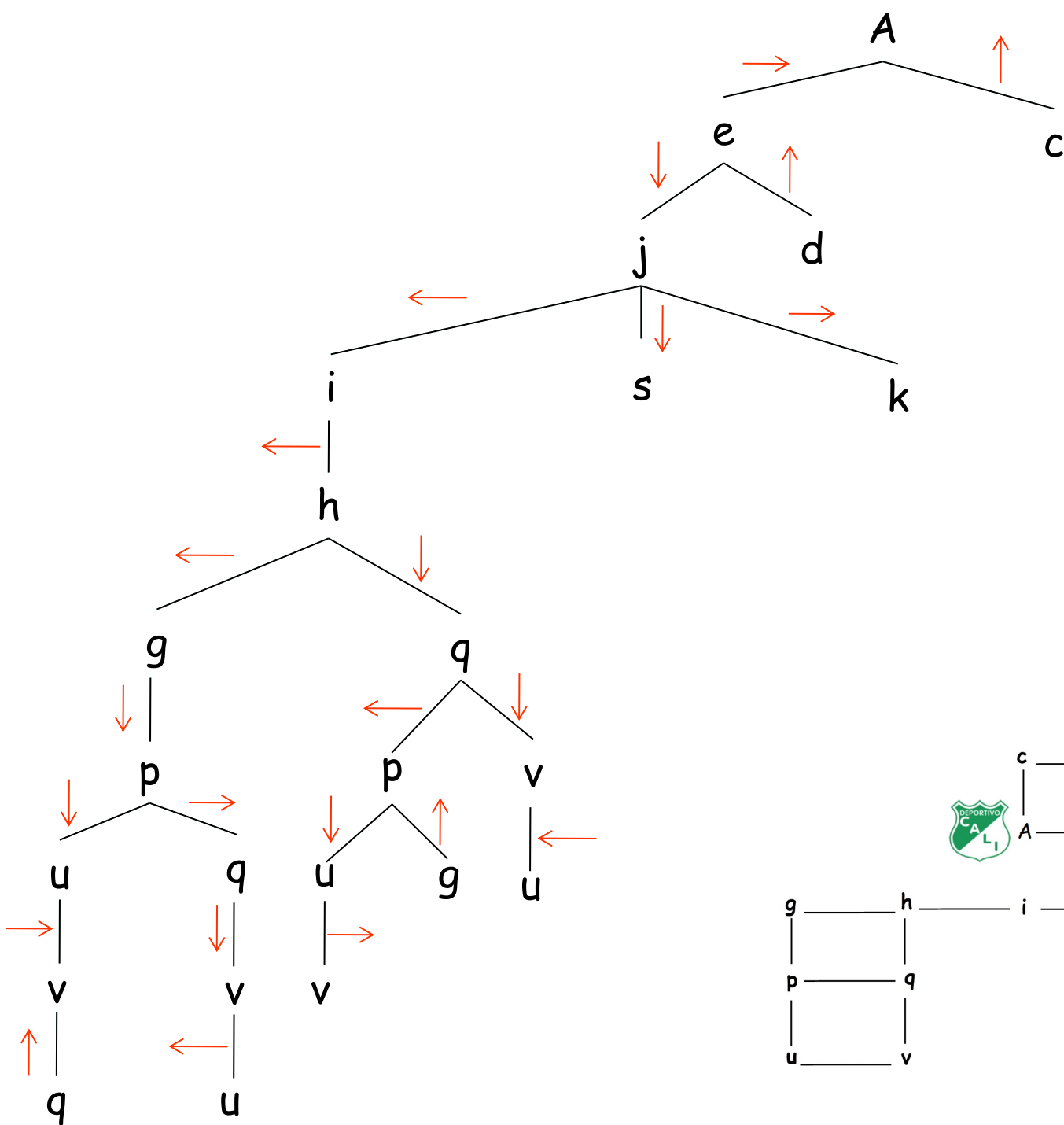


←, ↓, →, ↑

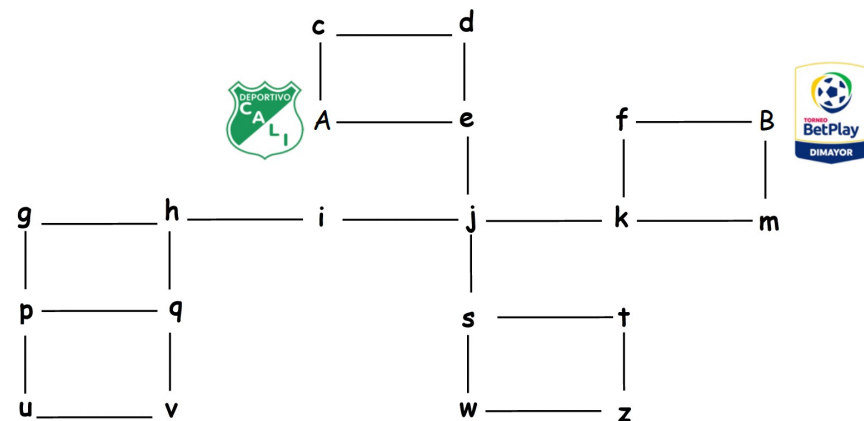


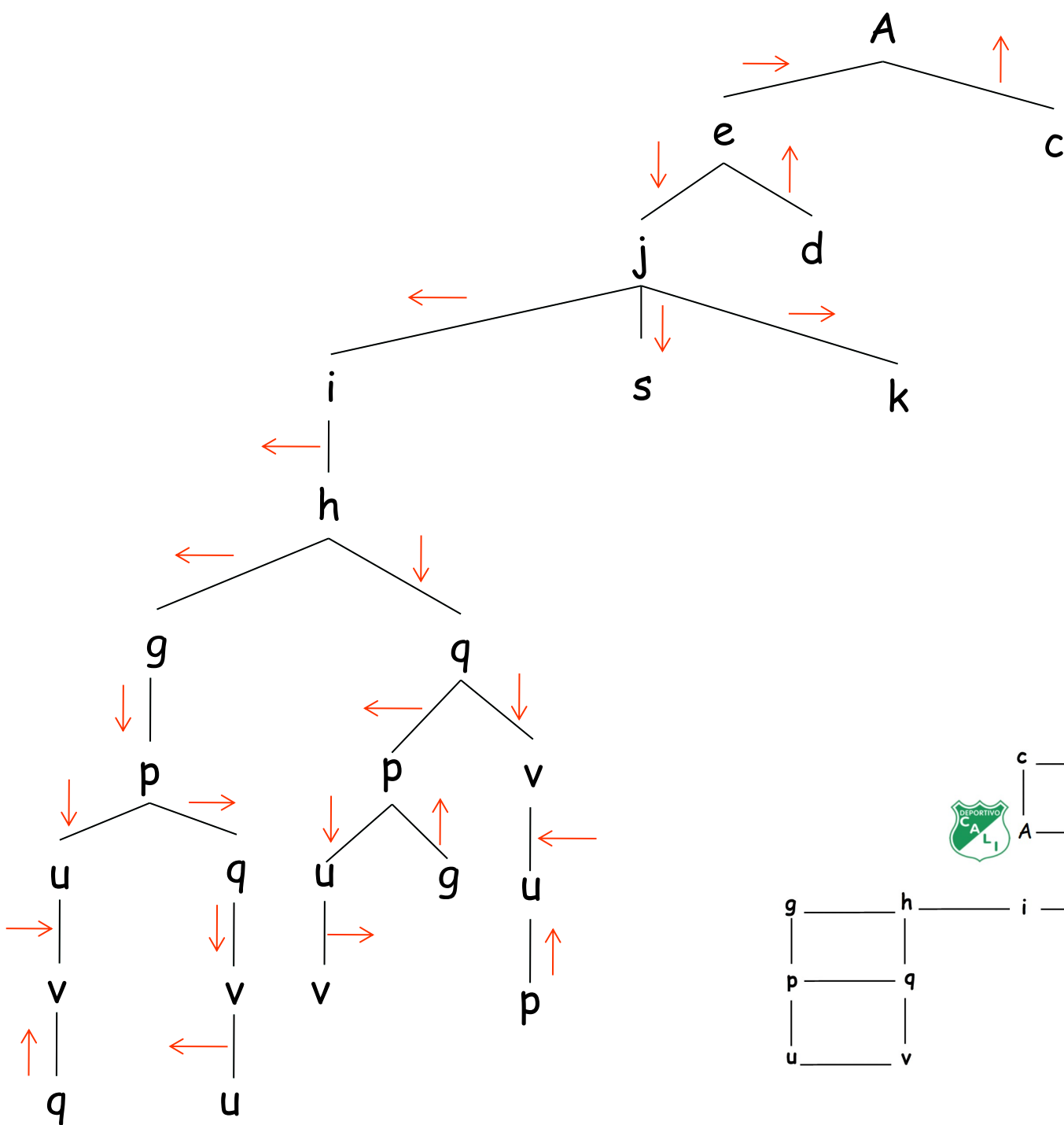




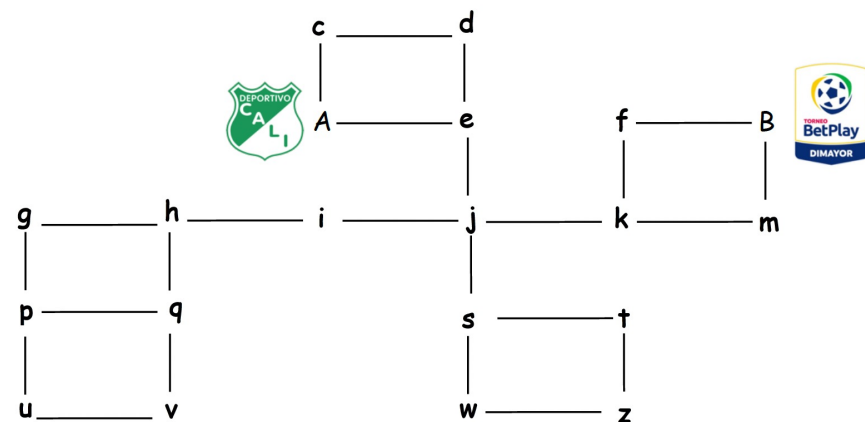


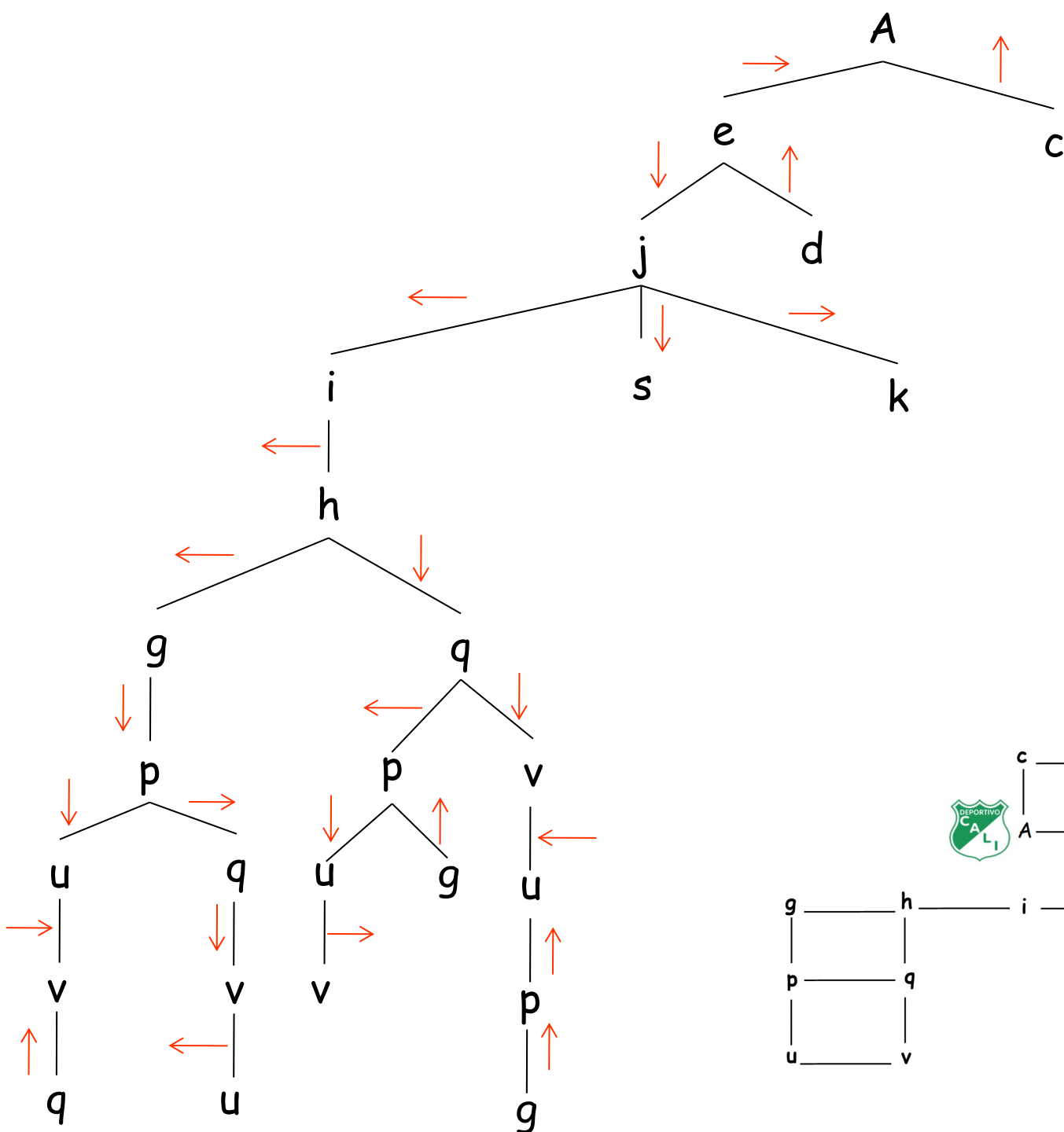
←, ↓, →, ↑



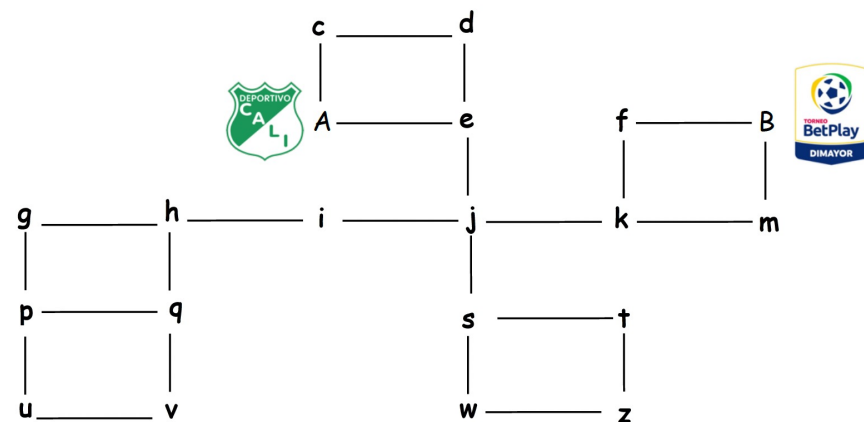


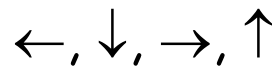
←, ↓, →, ↑

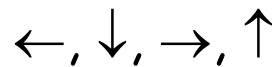


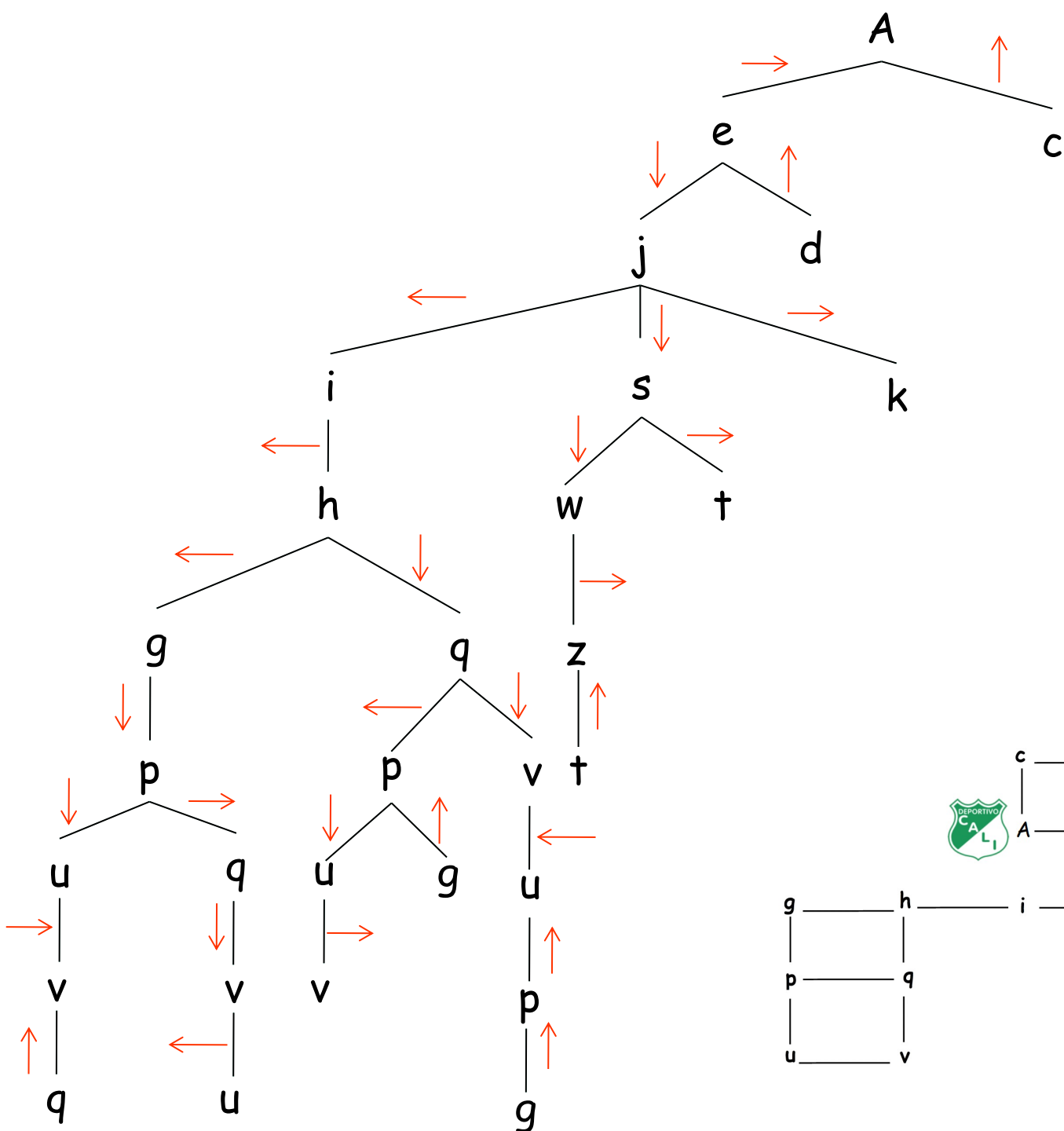


←, ↓, →, ↑

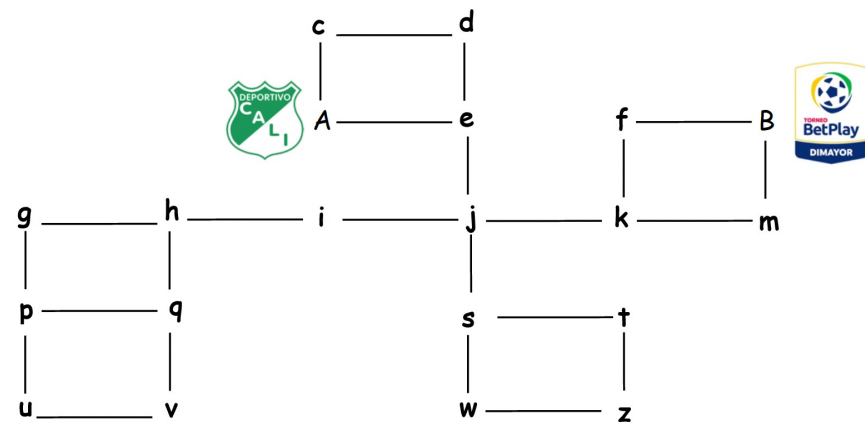


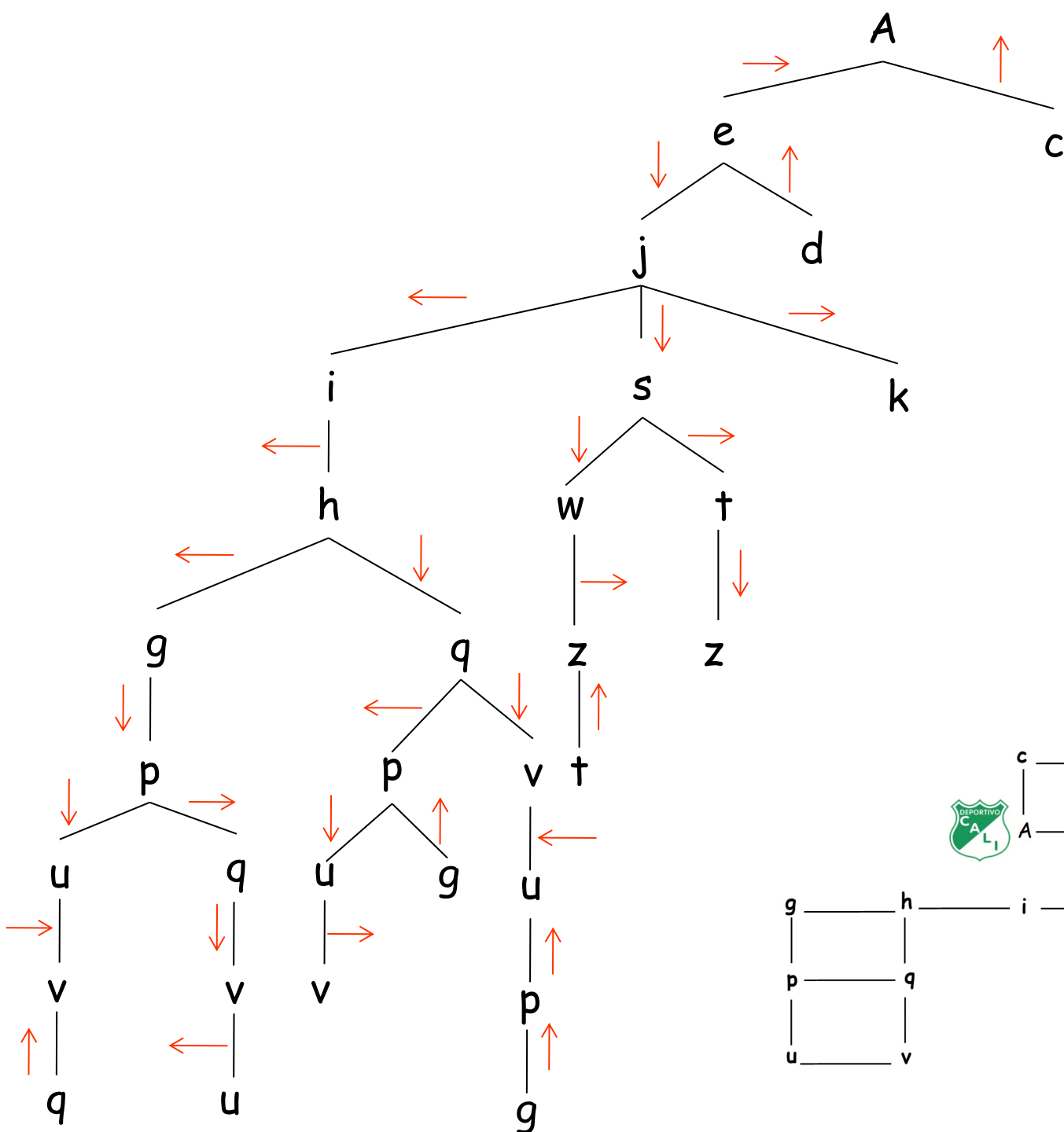




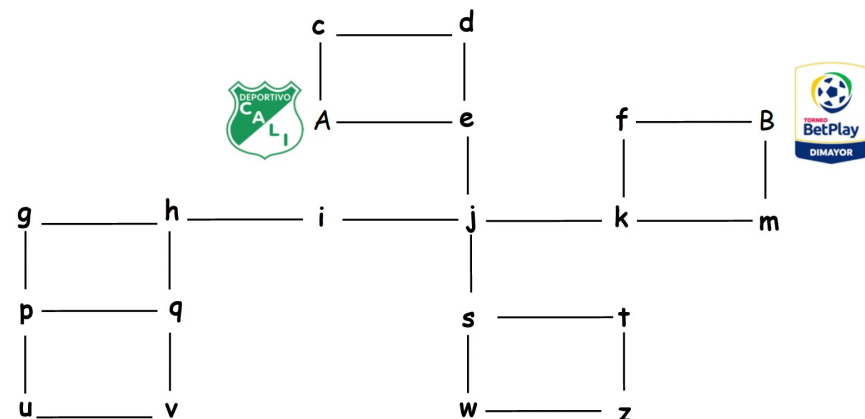


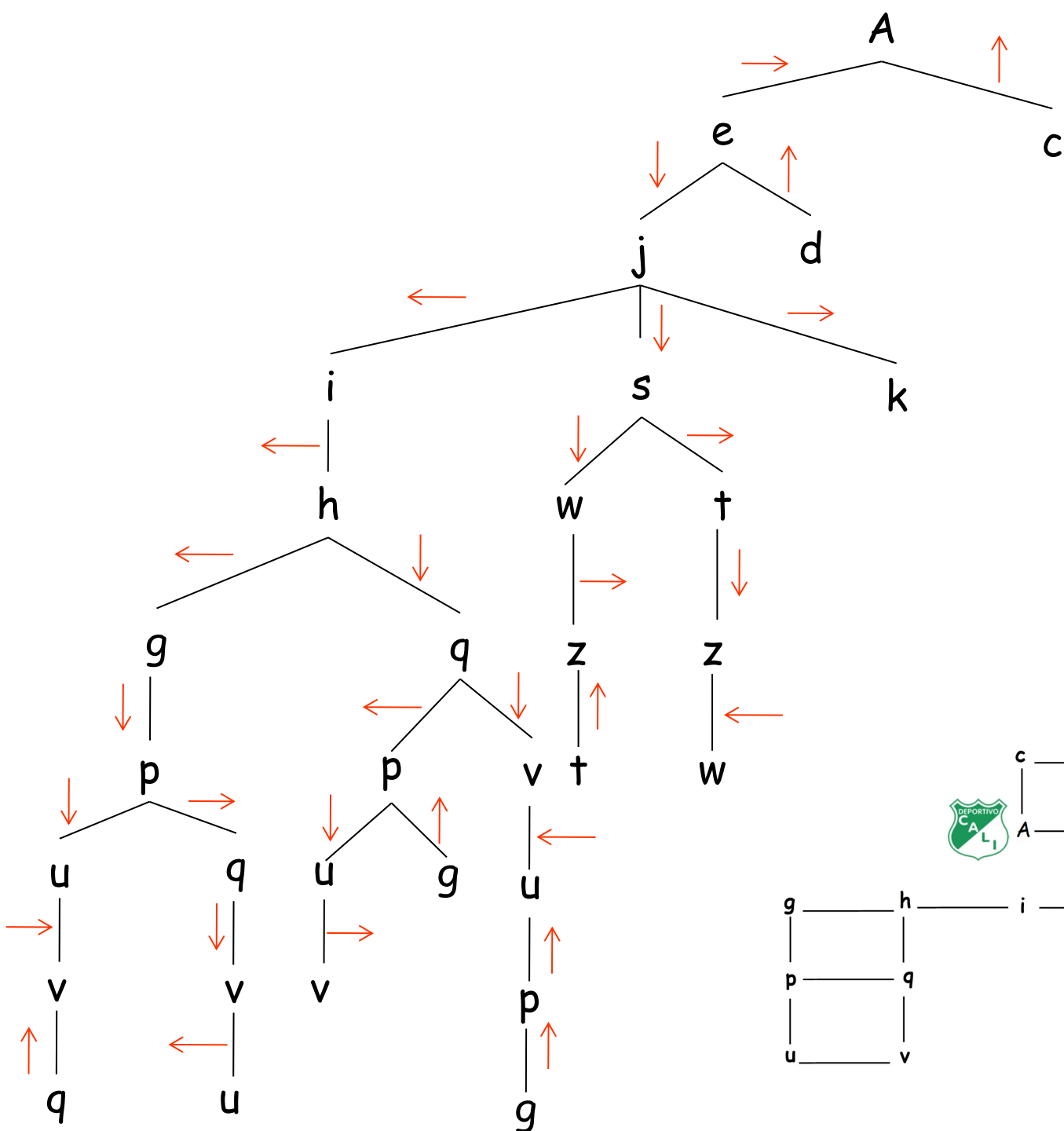
←, ↓, →, ↑



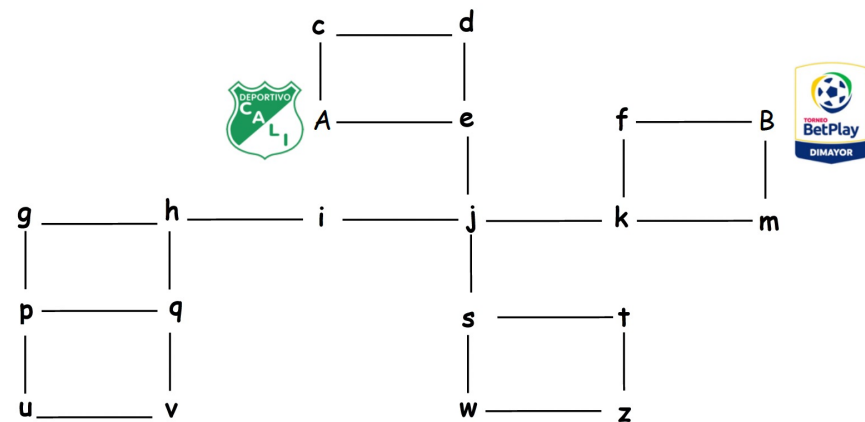


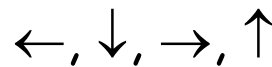
←, ↓, →, ↑

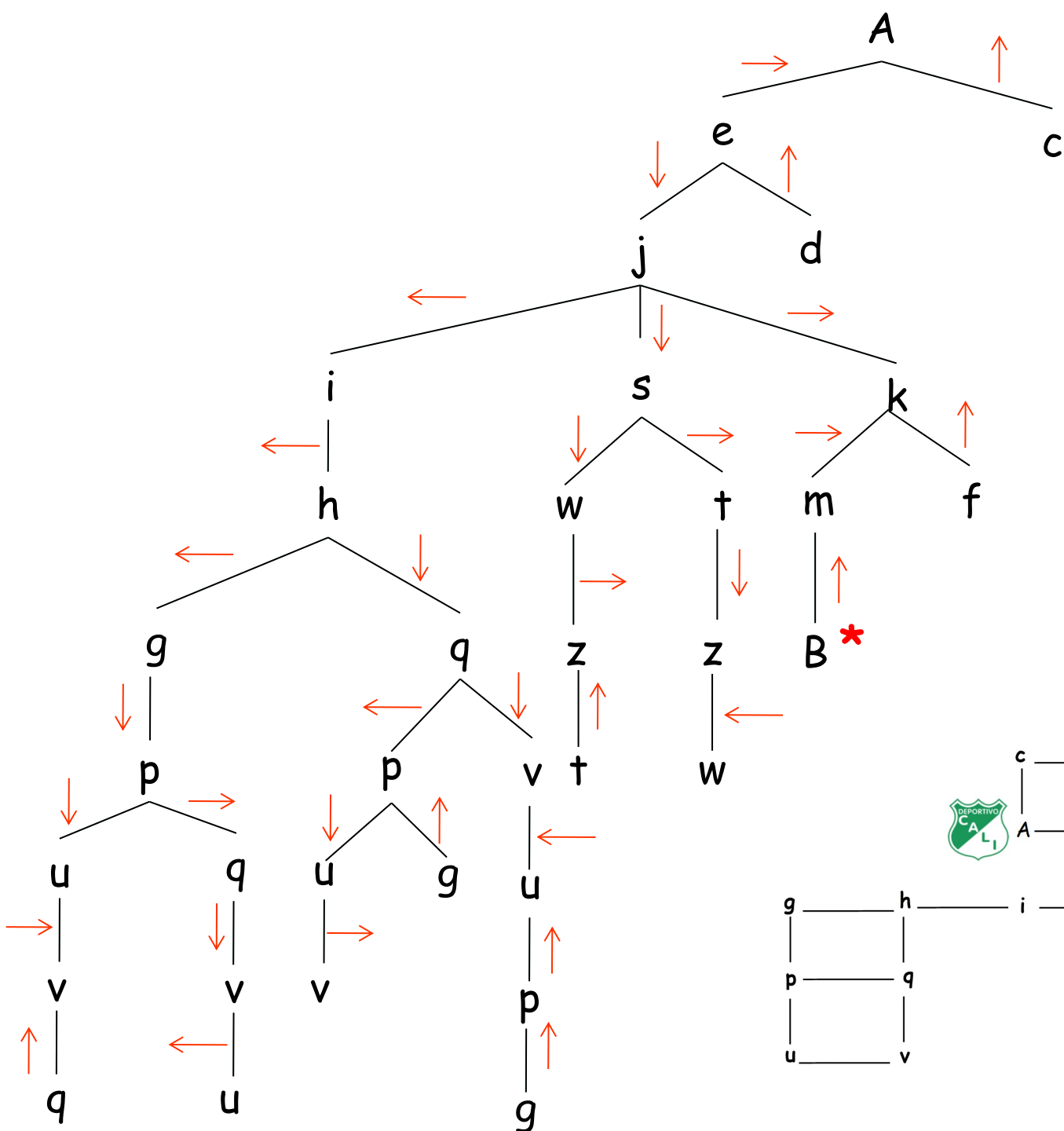




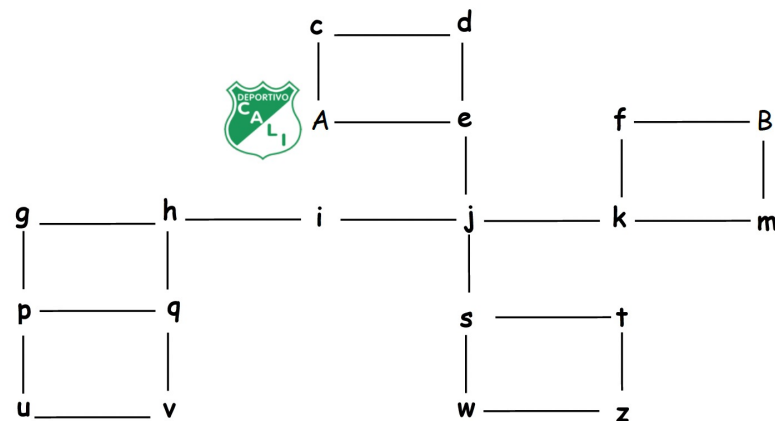
←, ↓, →, ↑







←, ↓, →, ↑



Búsqueda no informada

Búsqueda no informada:

- Preferente por amplitud
- De costo uniforme
- Preferente por profundidad
 - Preferente por profundidad evitando ciclos
- Limitada por profundidad
- Por profundización iterativa

Búsqueda no informada

Aplicar búsqueda preferente por profundidad evitando ciclos

Agente Minero. Indique el camino que sigue el agente.

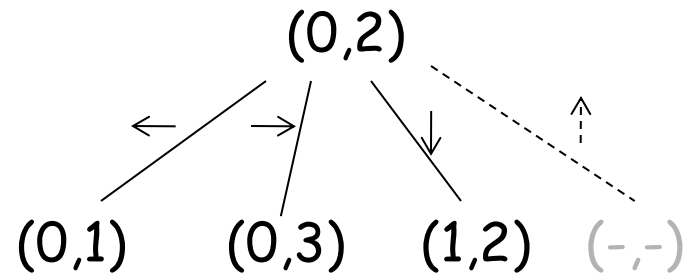


Punto de inicio



Lingote

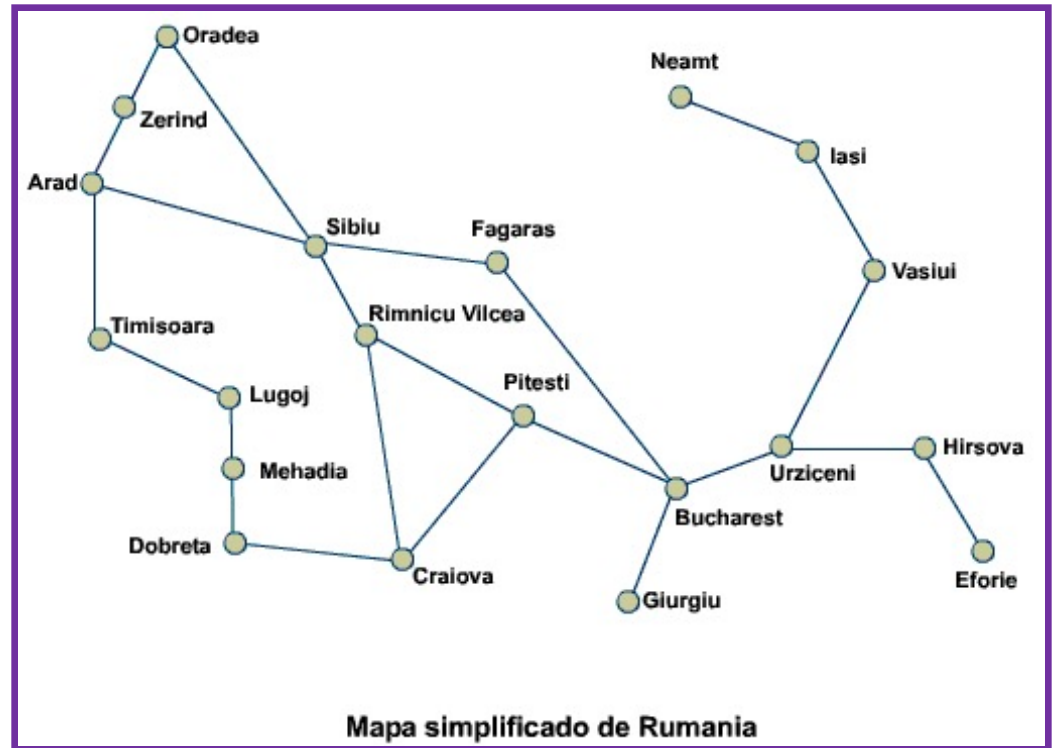
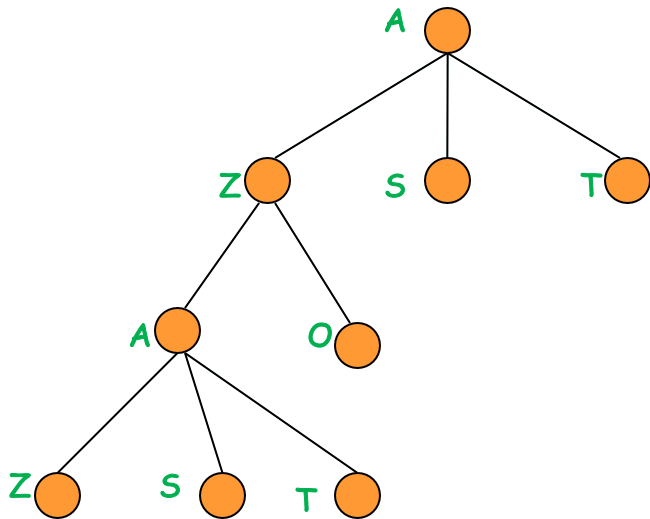
	0	1	2	3	4
0					
1					
2					
3					
4					



Considere el orden de los operadores como se muestra en el árbol

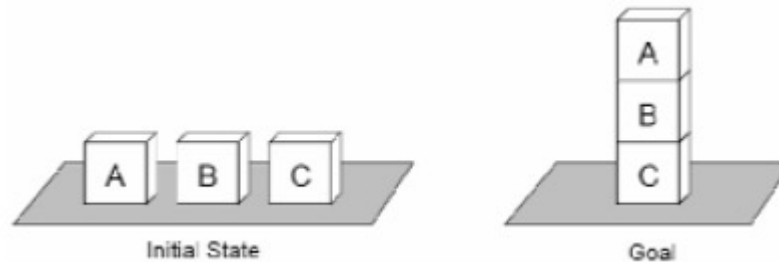
Búsqueda no informada

Ejemplo de bucle infinito



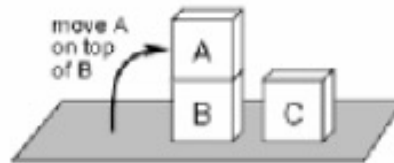
Búsqueda no informada

- El mundo de los bloques



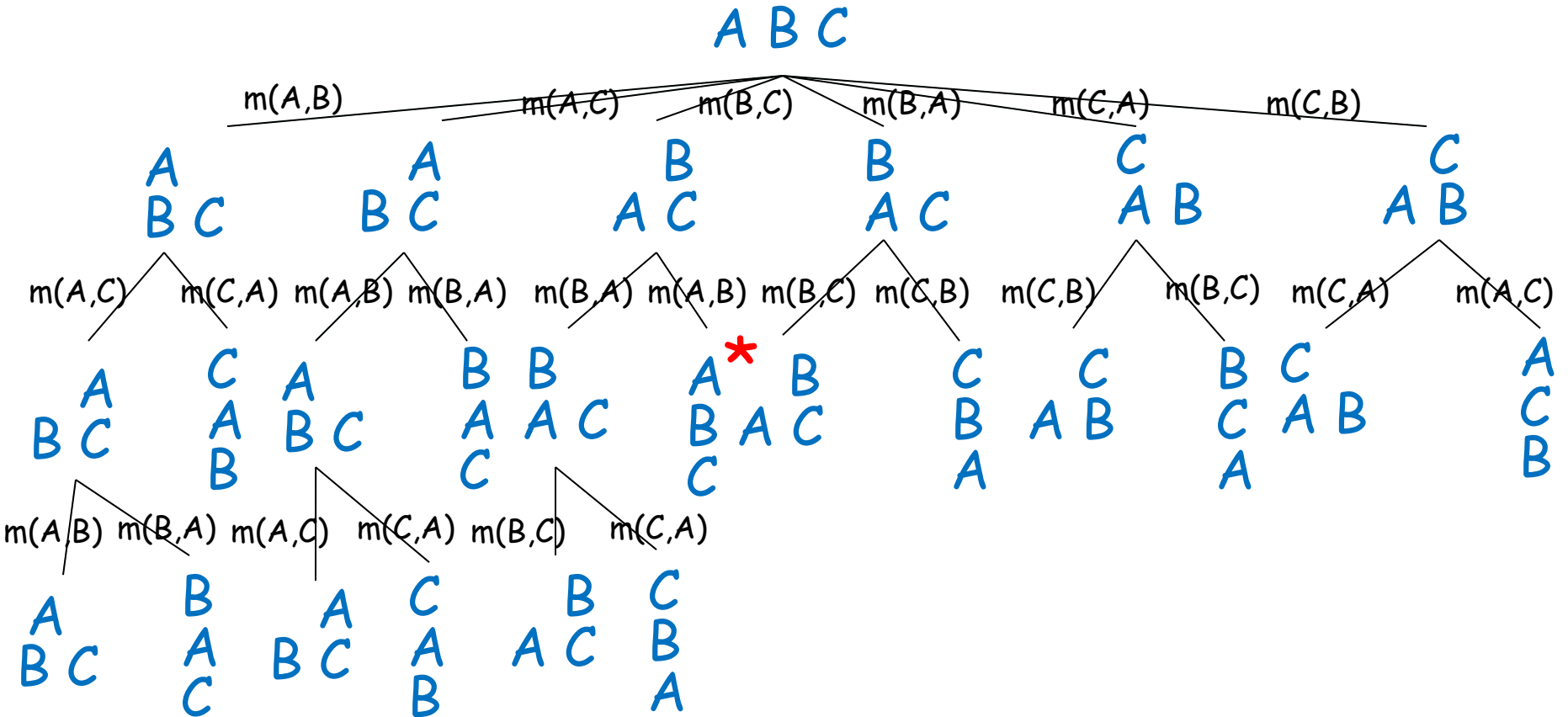
Operadores

$m(A, B)$: mover el bloque A encima del B . A puede estar sobre otro bloque.

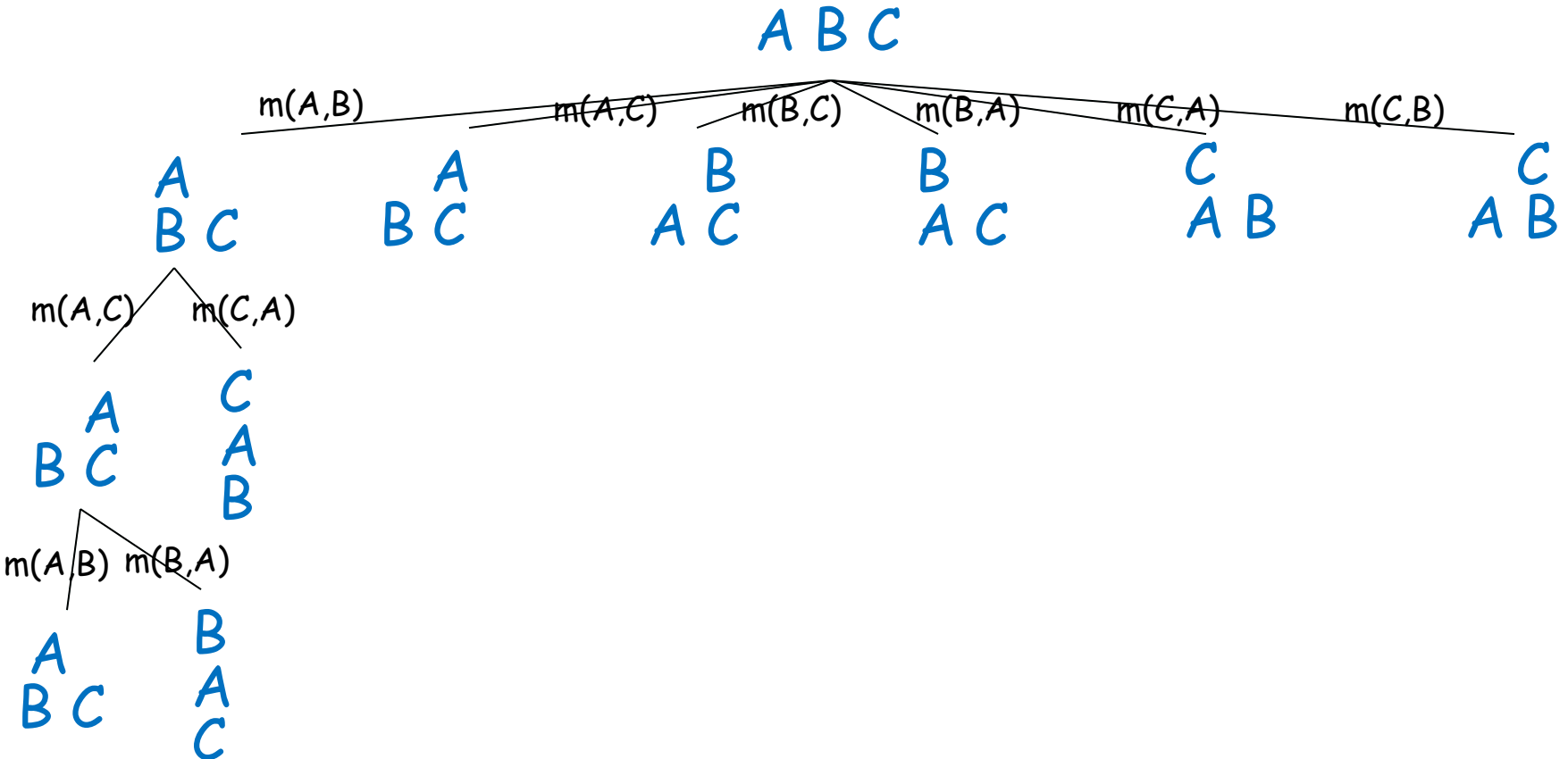


Aplique búsqueda preferente por amplitud y por profundidad sin evitar ciclos

- Búsqueda preferente por amplitud



- Búsqueda preferente por profundidad



* Se queda en un ciclo

Búsqueda no informada

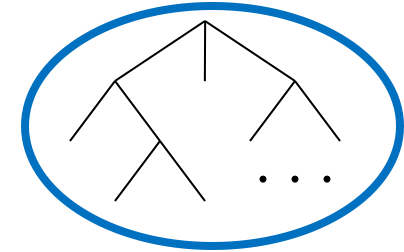
Búsqueda no informada:

- Preferente por amplitud
- De costo uniforme
- Preferente por profundidad
- Limitada por profundidad
- Por profundización iterativa

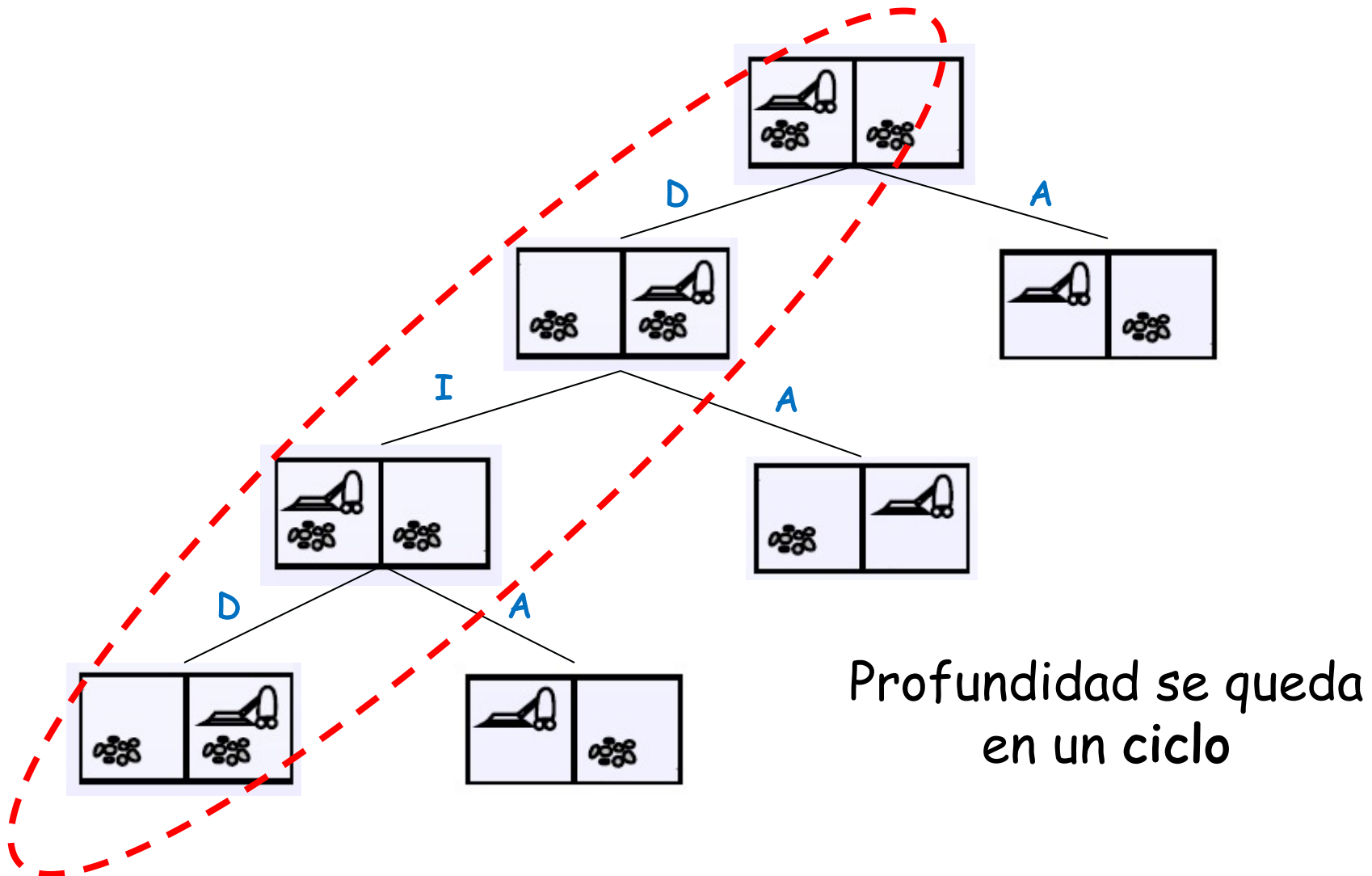
Búsqueda no informada

Problema: El mundo de la aspiradora

- Estados
- Operadores
- Prueba de meta
- Costo de ruta



Búsqueda no informada



Búsqueda no informada

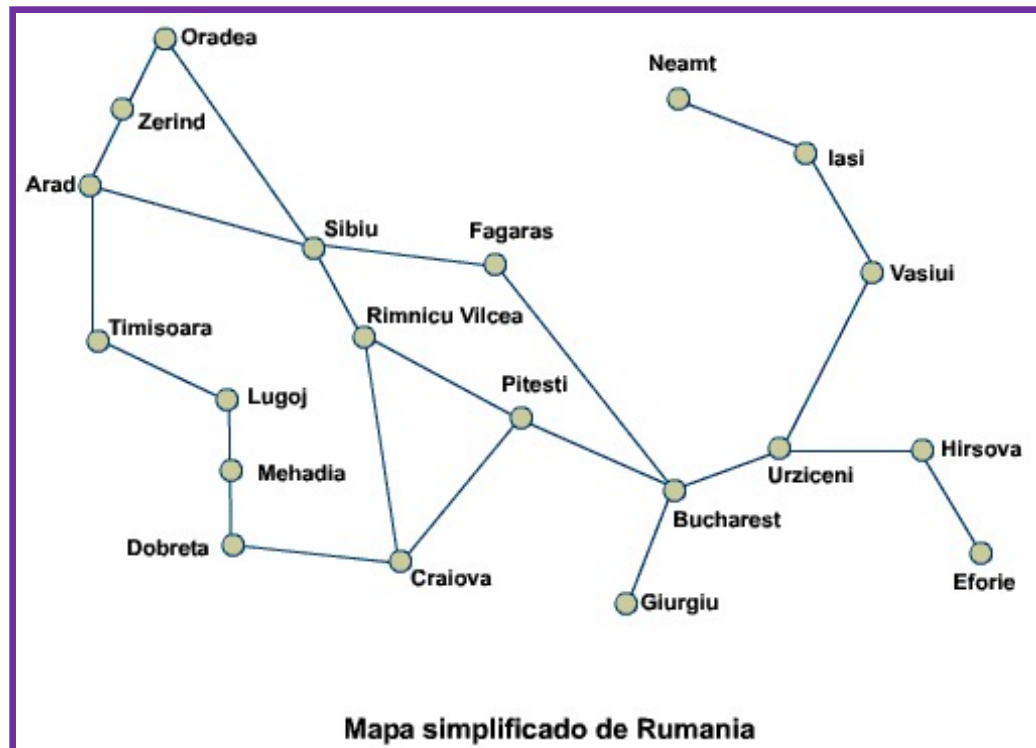
Búsqueda limitada por profundidad

- Se establece una **profundidad máxima** hasta la que se explora, cuando se llegue a ésta, se continúa con las profundidades anteriores
- El límite a usar depende del problema particular

Búsqueda no informada

Búsqueda limitada por profundidad

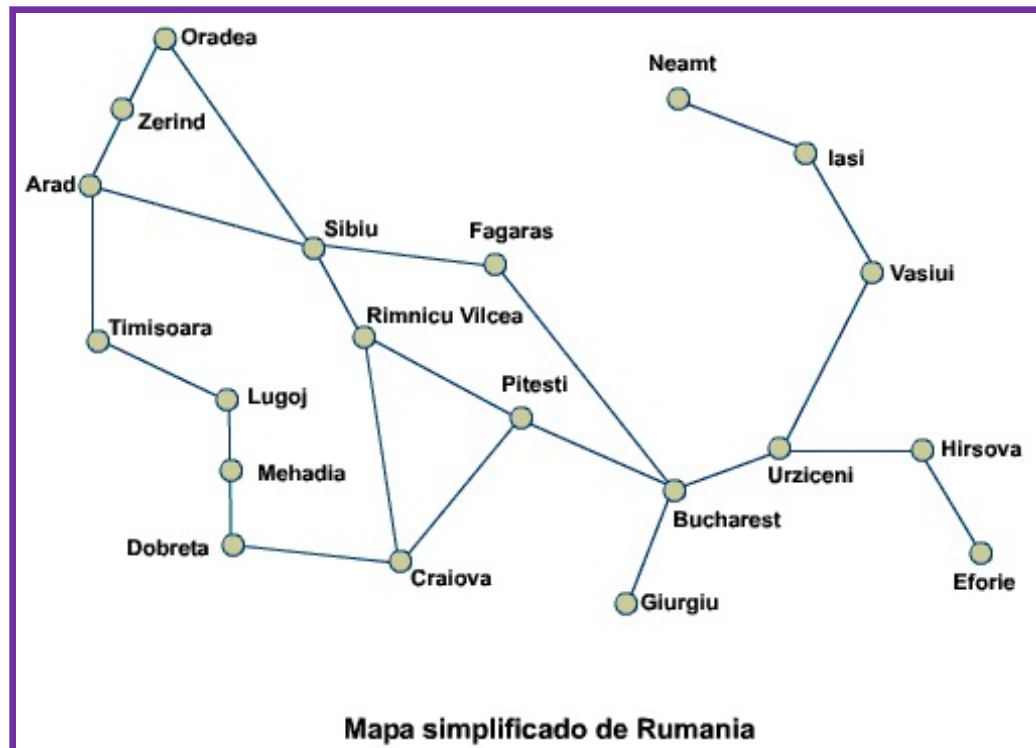
- ¿Cuál es un buen límite para este problema?



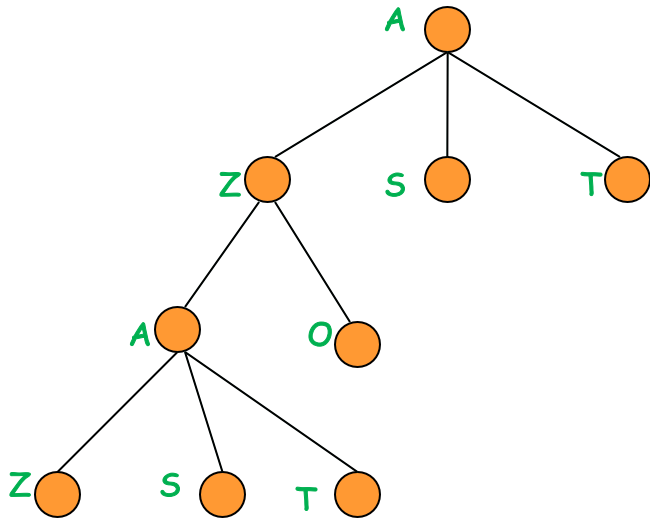
Búsqueda no informada

Búsqueda limitada por profundidad

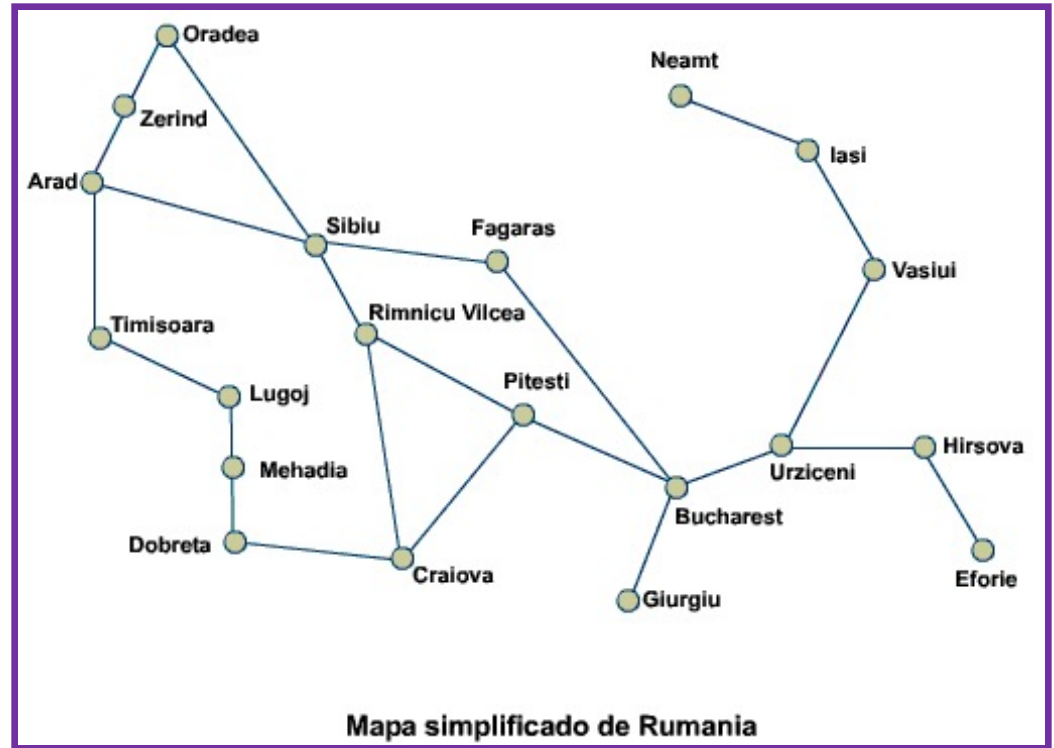
- Hay 20 ciudades, la solución tendrá máximo 19 ciudades



Búsqueda no informada



El ciclo se detiene en la profundidad máxima



Búsqueda no informada

¿Qué límite utilizaría
para la profundidad de
este problema?

6	8	2
	5	3
7	4	1

Búsqueda no informada

¿Qué límite utilizaría para la profundidad de este problema?

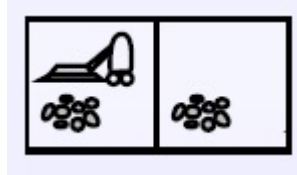
6	8	2
	5	3
7	4	1

- En muchos problemas no es fácil saber el límite
- ¿Qué pasaría si se selecciona un límite que es menor a la profundidad donde se encuentra la solución?

Búsqueda no informada

Aplicar búsqueda limitada
por profundidad.

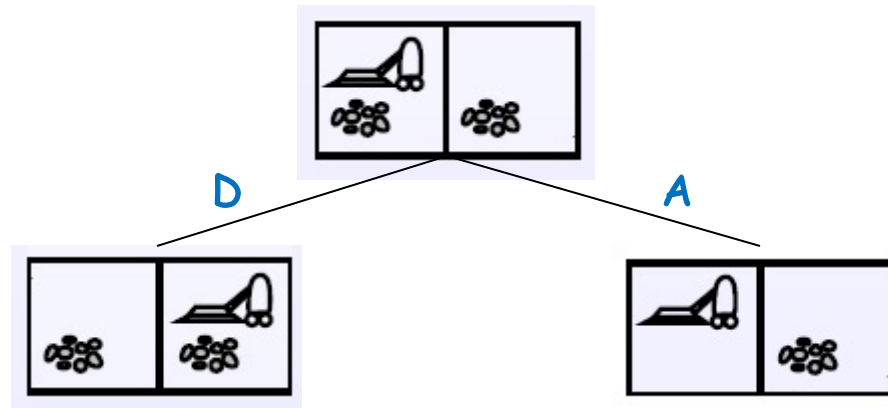
- Profundidad límite: 4

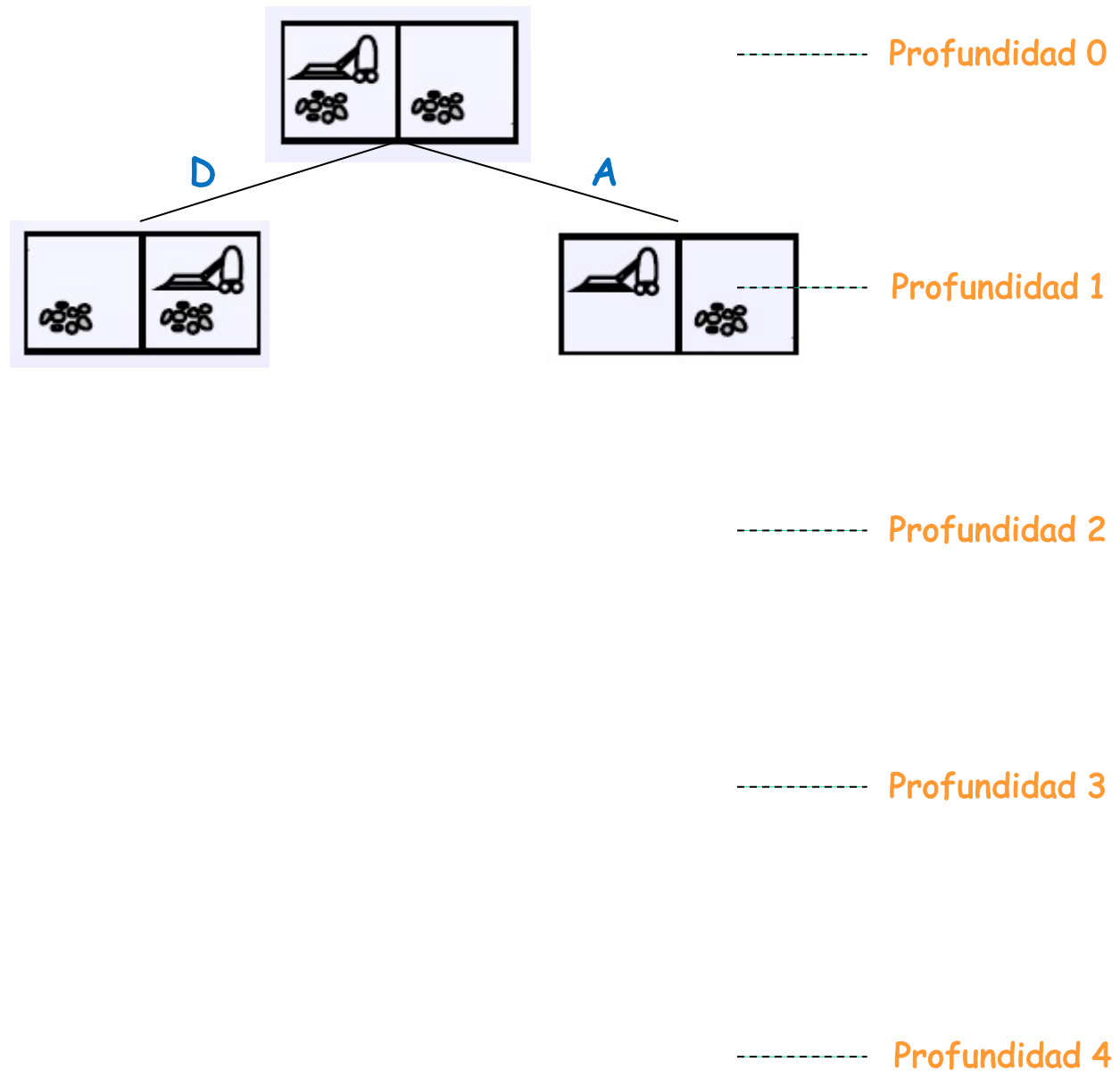


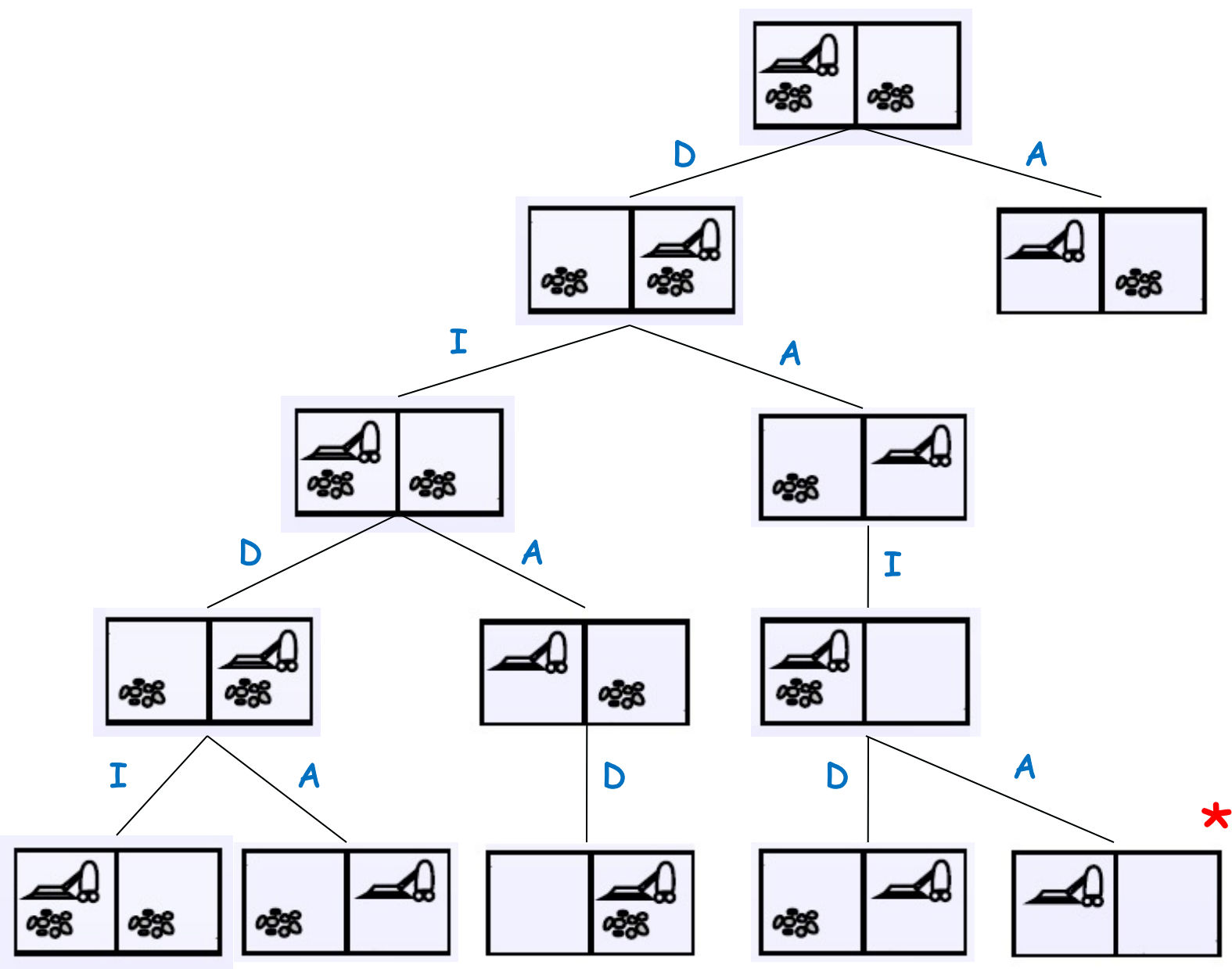
Aplique los operadores en el orden:

- Derecha
- Izquierda
- Aspirar

* No evite ni ciclos ni devolverse







Búsqueda no informada

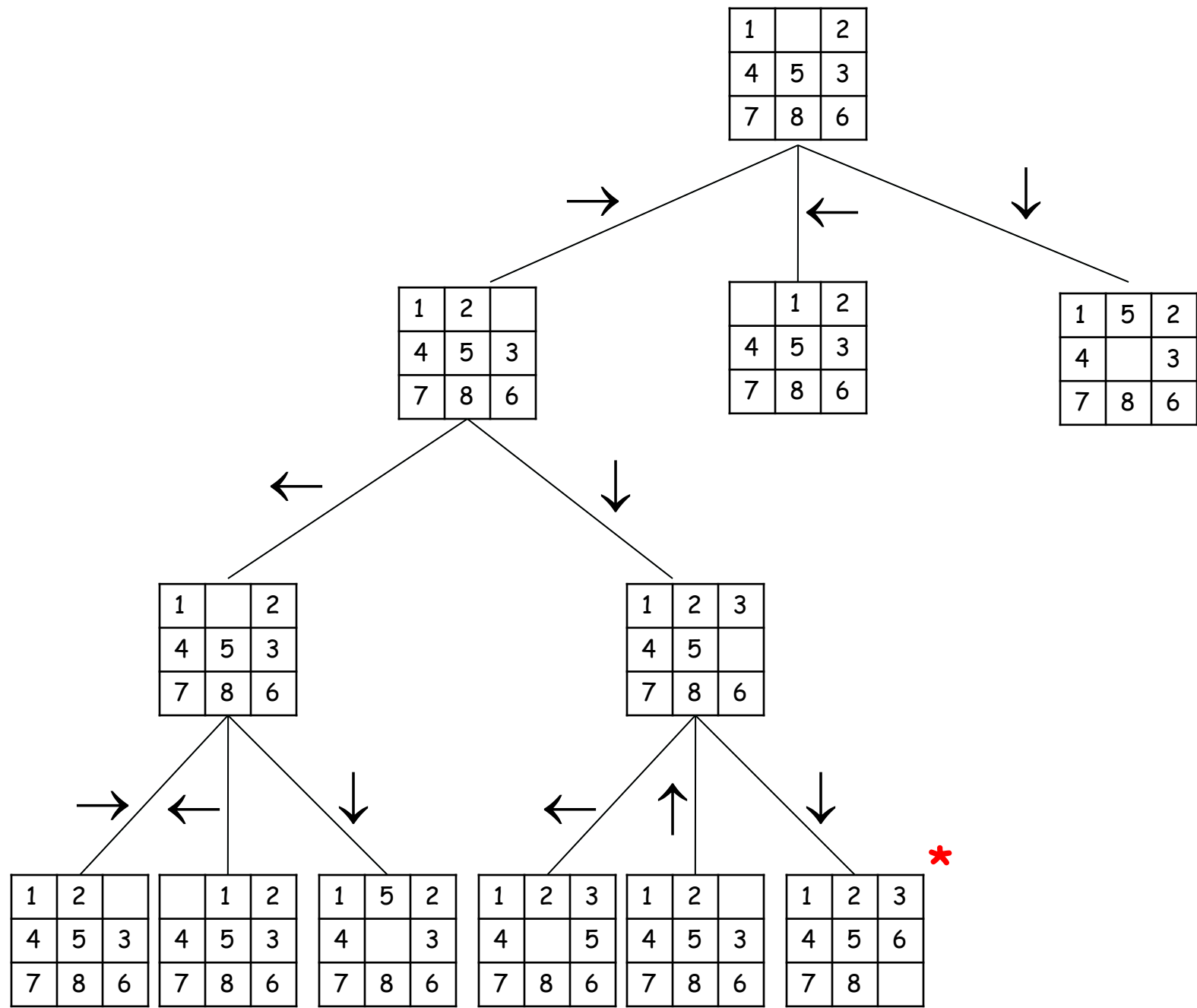
Aplicar búsqueda limitada por profundidad.

-Profundidad límite: 3

1		2
4	5	3
7	8	6

Aplique los operadores en el orden: $\rightarrow, \leftarrow, \uparrow, \downarrow$

* No evite ni devolverse ni ciclos



Búsqueda no informada

Búsqueda limitada por profundidad

- Es completa (siempre y cuando el límite seleccionado incluya alguna solución)
- Complejidad temporal $O(b^l)$, donde l es la profundidad máxima
- Complejidad espacial $O(b \cdot l)$, donde l es la profundidad máxima
- No es óptima

Búsqueda no informada

Búsqueda limitada por profundidad

- Intenta tomar lo bueno de preferente por profundidad que consiste en que se puede llegar más rápido a una solución que amplitud y se intenta solucionar el problema de los bucles infinitos

Búsqueda no informada

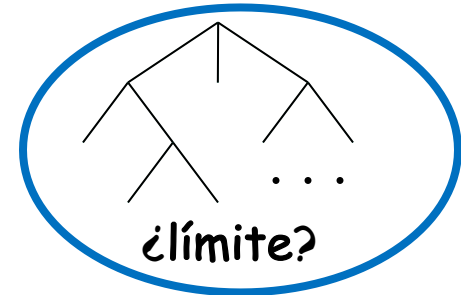
Búsqueda no informada:

- Preferente por amplitud
- De costo uniforme
- Preferente por profundidad
- Limitada por profundidad
- Por profundización iterativa

Búsqueda no informada

¿Qué límite utilizaría para la profundidad de este problema?

6	8	2
	5	3
7	4	1



Búsqueda no informada

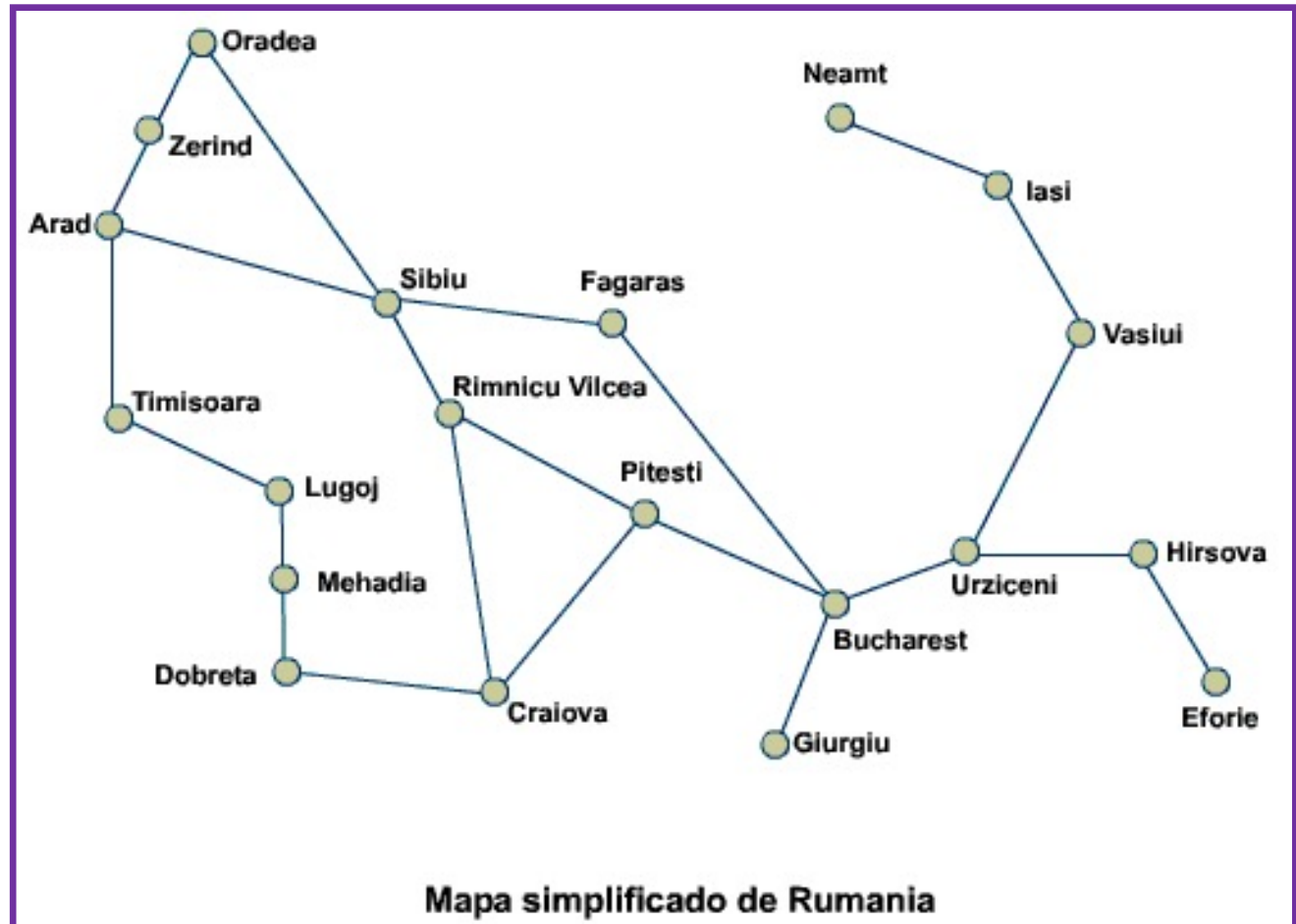
Búsqueda por profundización iterativa

- Prueba todos los límites de profundidad posibles, inicialmente toma como límite 0 y lleva a cabo búsqueda limitada por profundidad, luego aumenta el límite en 1 y realiza el mismo procedimiento, así hasta encontrar alguna solución

Búsqueda no informada

Aplicar búsqueda por profundización iterativa para ir de Arad a Bucarest

Evite devolverse



Búsqueda no informada

• Límite 0:

Arad

Búsqueda no informada

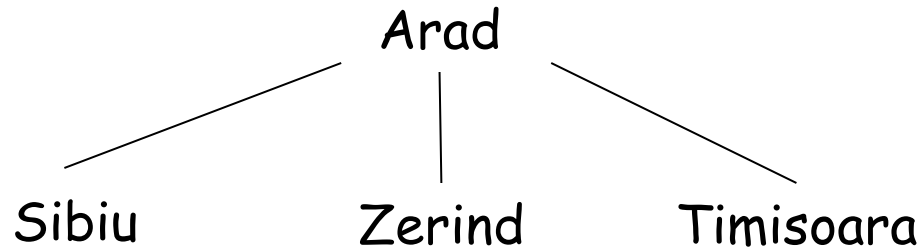
- Límite 0: Arad
- Límite 1: Arad

Búsqueda no informada

- Límite 0:

Arad

- Límite 1:

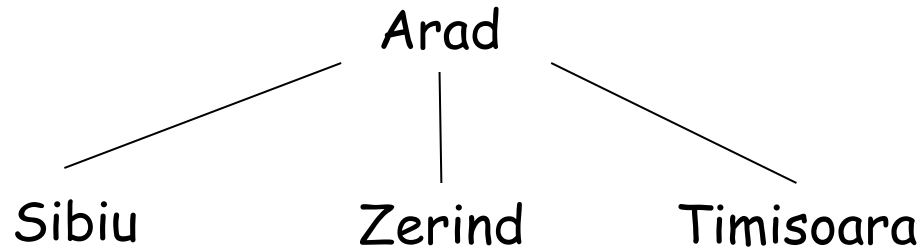


Búsqueda no informada

- Límite 0:

Arad

- Límite 1:



- Límite 2:

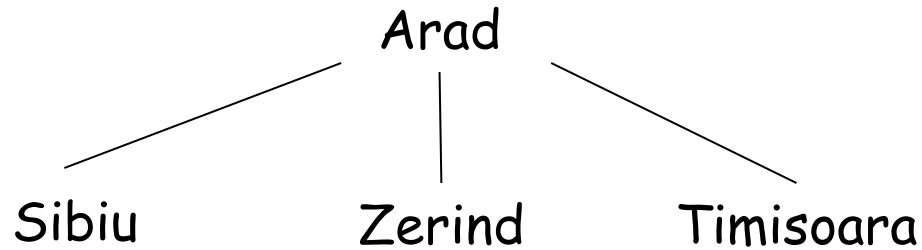
Arad

Búsqueda no informada

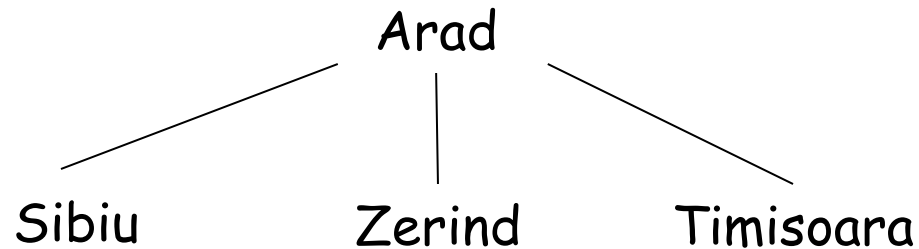
- Límite 0:

Arad

- Límite 1:



- Límite 2:

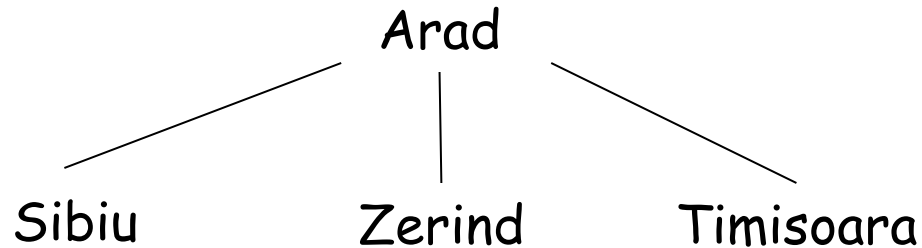


Búsqueda no informada

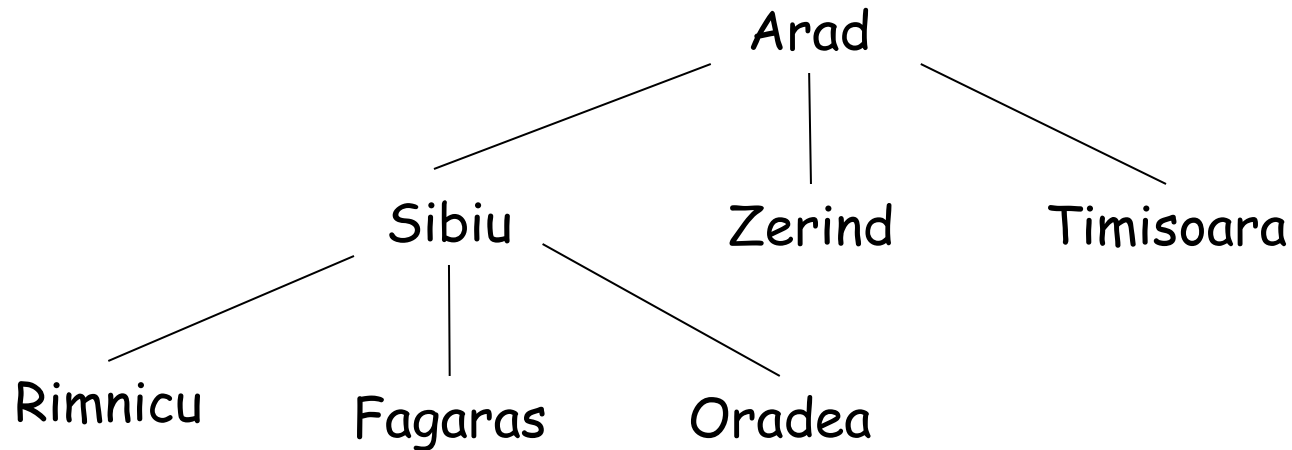
- Límite 0:

Arad

- Límite 1:



- Límite 2:

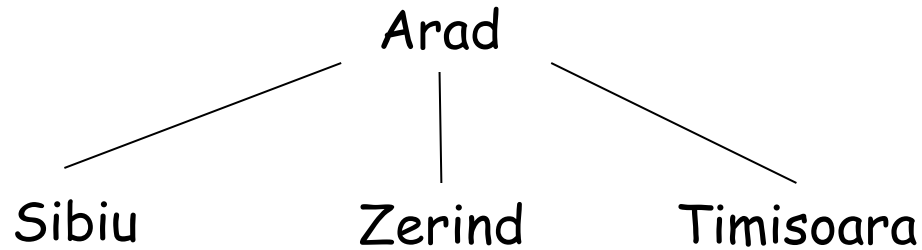


Búsqueda no informada

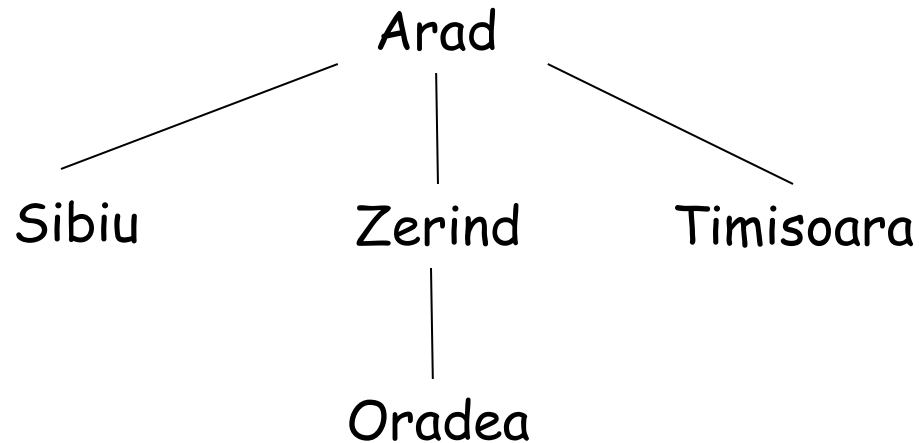
- Límite 0:

Arad

- Límite 1:



- Límite 2:

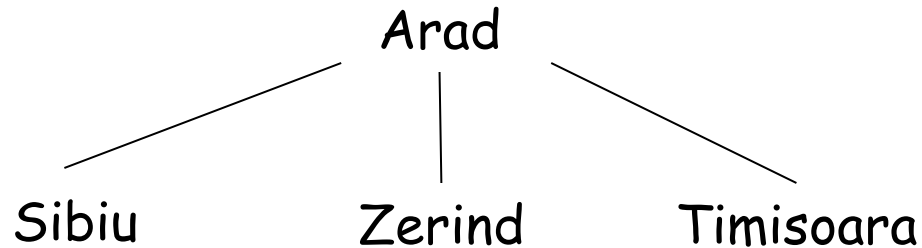


Búsqueda no informada

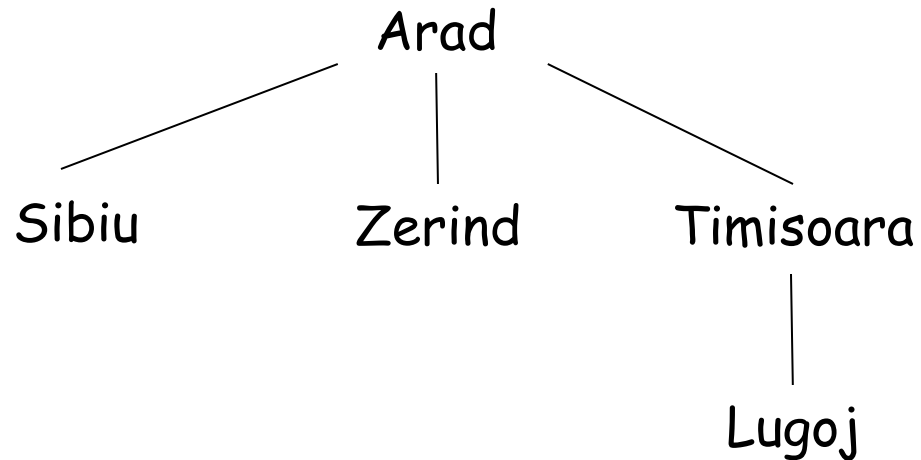
- Límite 0:

Arad

- Límite 1:



- Límite 2:



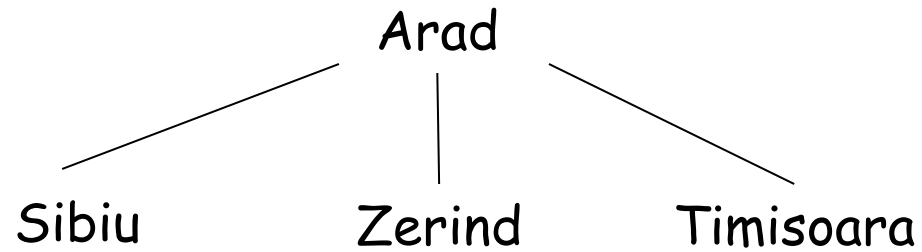
Búsqueda no informada

- Límite 3:

Arad

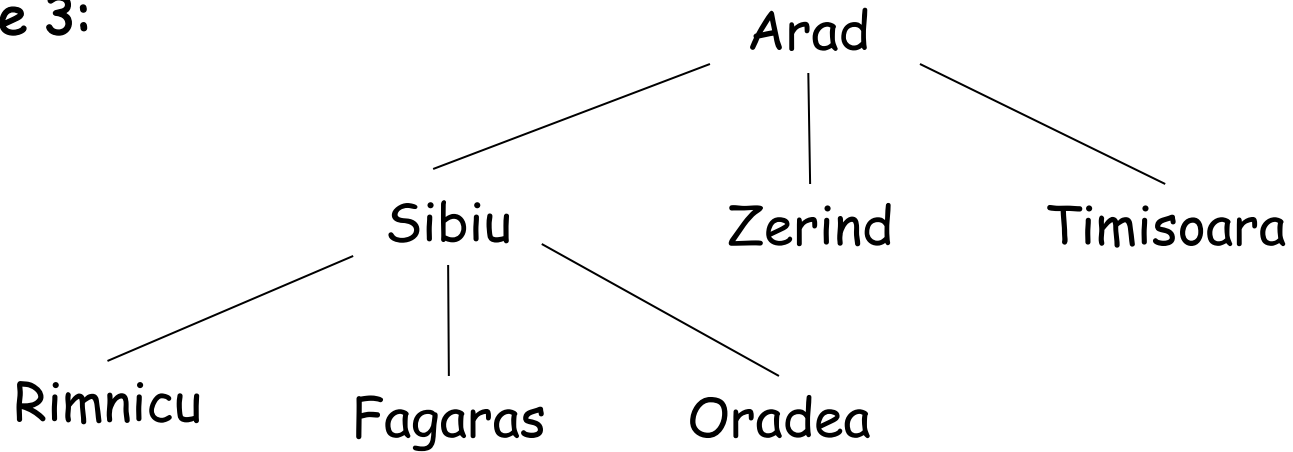
Búsqueda no informada

- Límite 3:



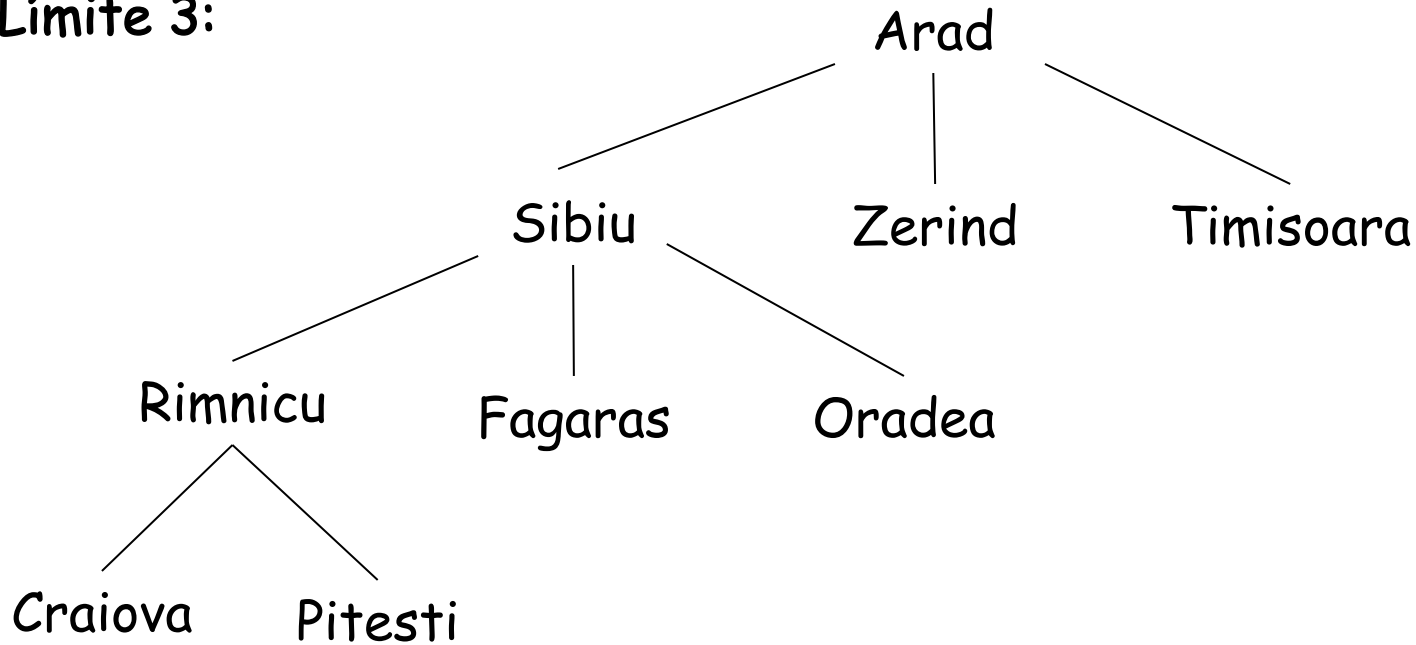
Búsqueda no informada

- Límite 3:



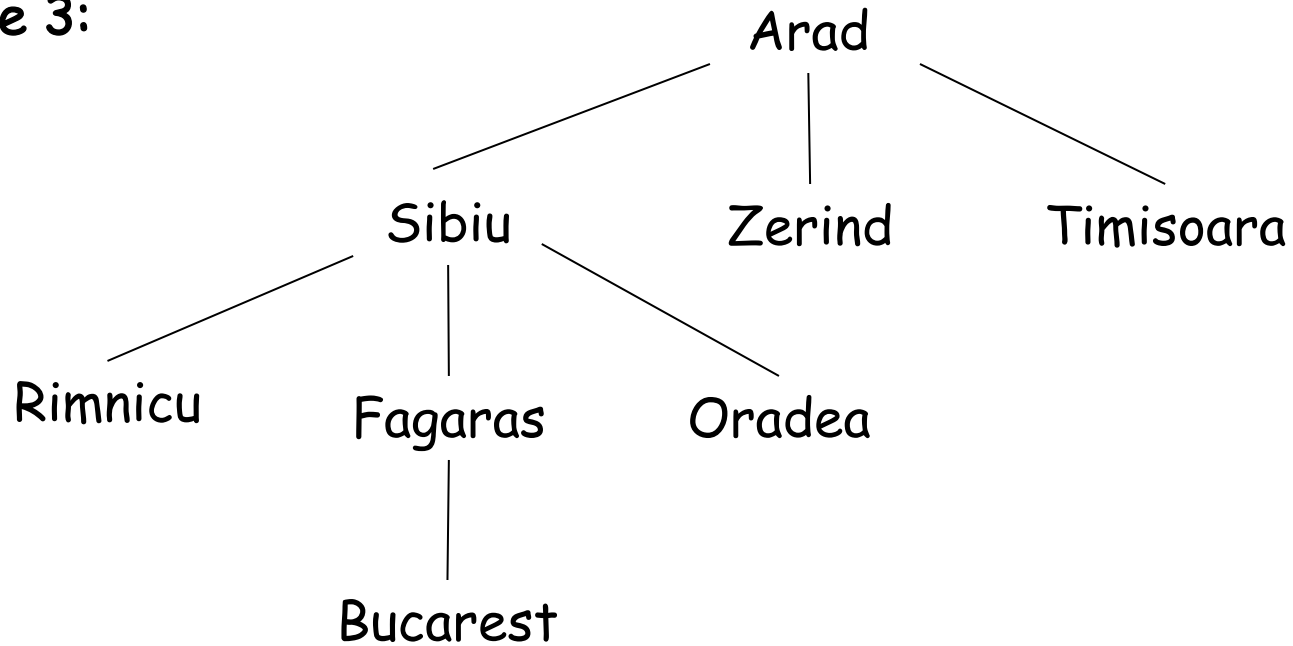
Búsqueda no informada

- Límite 3:



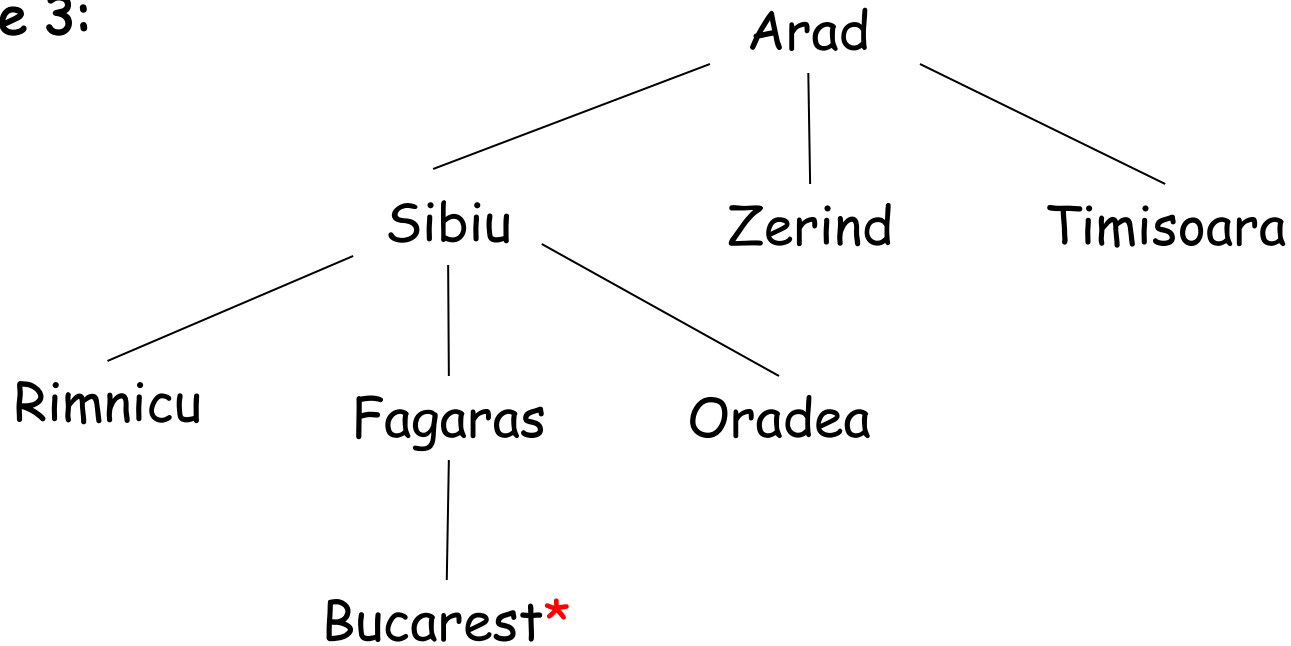
Búsqueda no informada

- Límite 3:



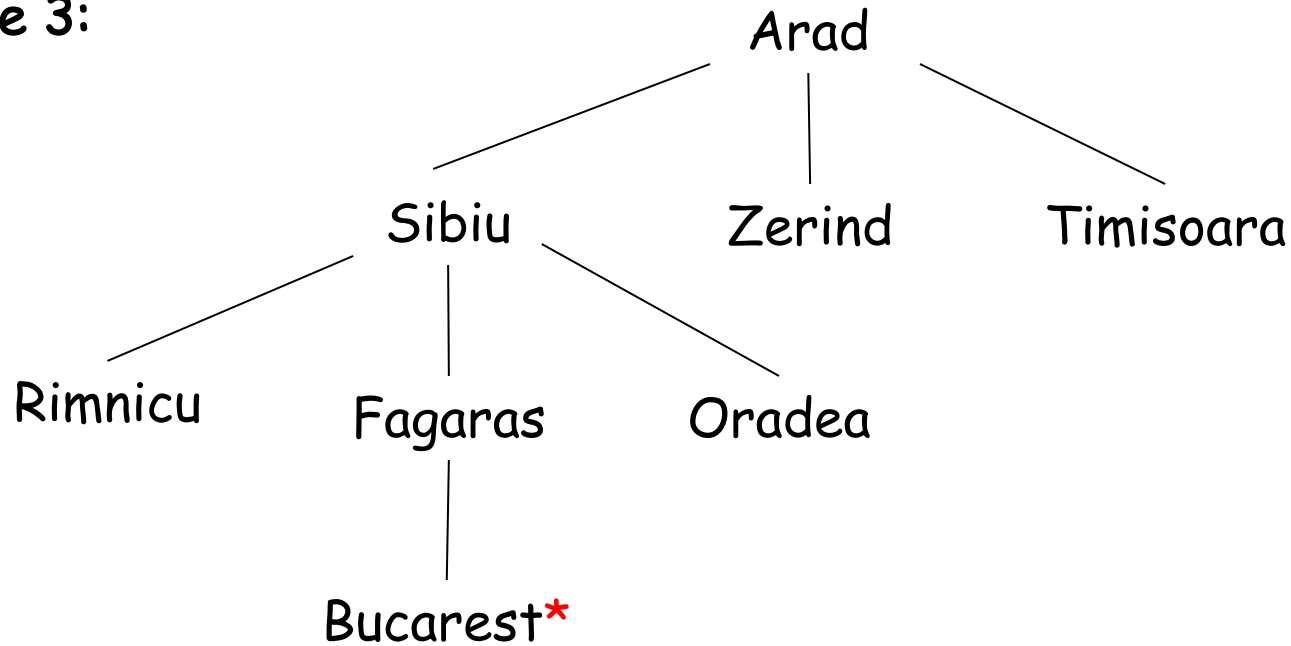
Búsqueda no informada

- Límite 3:



Búsqueda no informada

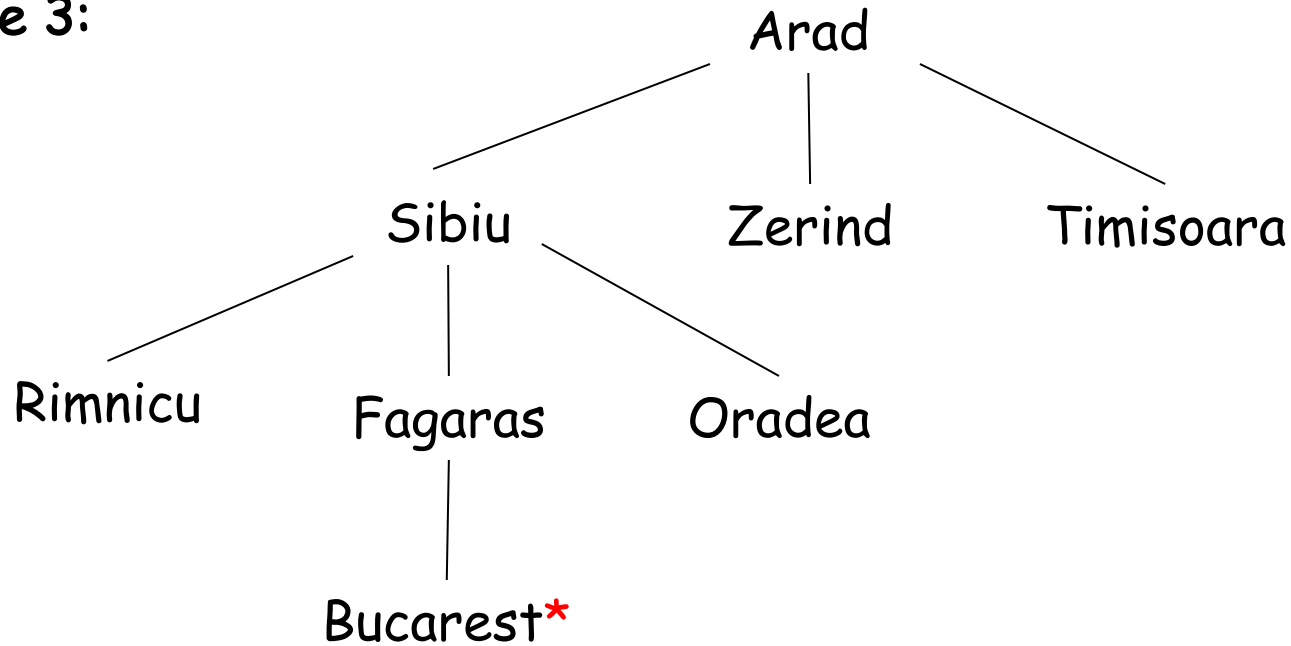
- Límite 3:



Veces expandida				
Profundidad	0, Arad	1. (Sibiu,Zerind,Timisoara)	2. (Rimnicu,Fagaras,Oradea), (Oradea), (Lugoj)	3.(Craiova,Pitesti),(Bucarest) ...

Búsqueda no informada

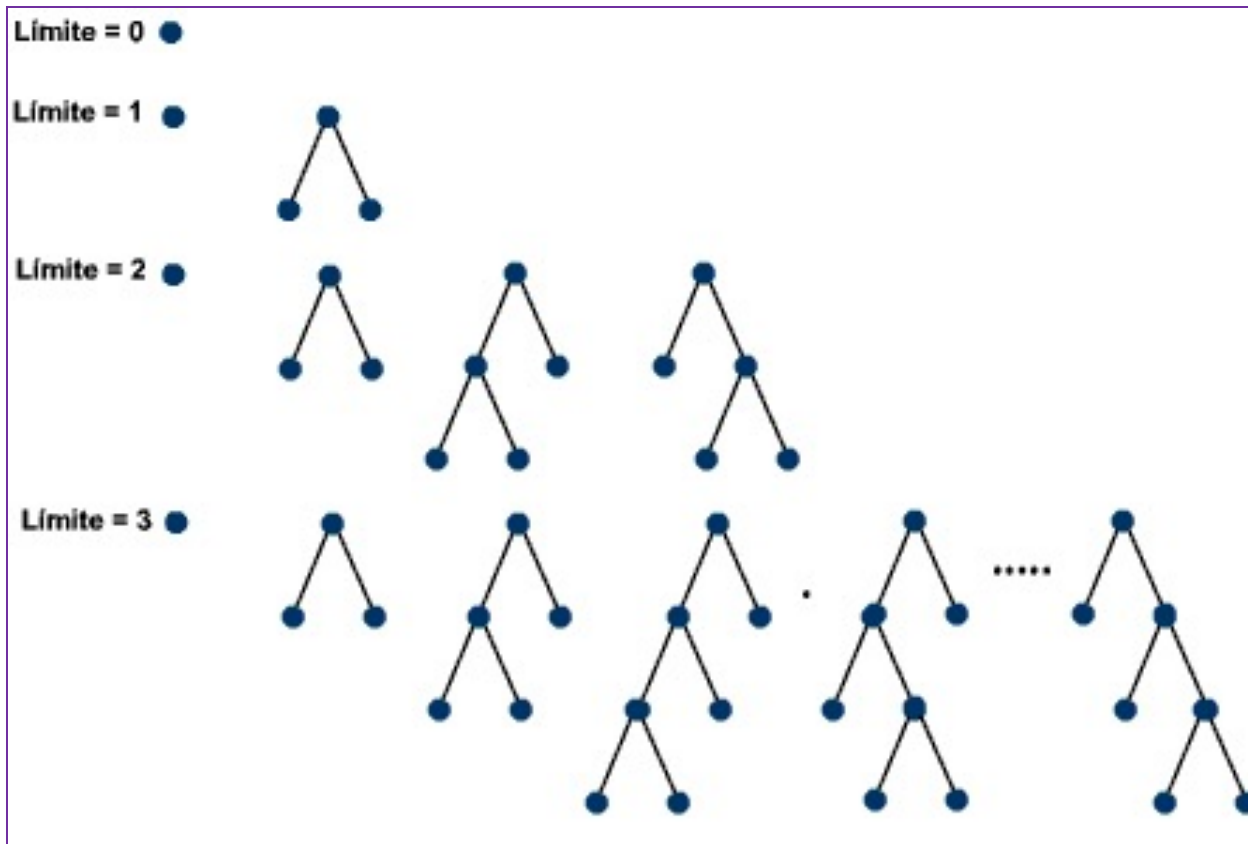
- Límite 3:



Veces expandida	4	3	2	1
Profundidad	0, Arad	1. (Sibiu,Zerind,Timisoara)	2. (Rimnicu,Fagaras,Oradea), (Oradea), (Lugoj)	3.(Craiova,Pitesti),(Bucarest)

Búsqueda no informada

Búsqueda por profundización iterativa



Búsqueda no informada

Búsqueda por profundización iterativa

- Note que hay nodos que se vuelven a expandir en cada nivel, por ejemplo, el nodo raíz se expande $(d+1)$ veces, sus b hijos d veces, sus hijos a su vez, que son b^2 , se expanden $d-1$ veces
- La complejidad temporal es:
$$(d+1)*1 + (d)*b + (d-1)*b^2 + \dots + 1*b^d = O(b^d)$$
- La complejidad espacial sigue siendo $O(b*d)$

Búsqueda no informada

Búsqueda no informada:

- Preferente por amplitud
- De costo uniforme
- Preferente por profundidad
- Limitada por profundidad
- Por profundización iterativa

Búsqueda no informada

Búsqueda no informada

- El ratón inteligente.



Considere un ratón que aplica algoritmos de búsqueda para desplazarse en el ambiente. El costo de cada movimiento es 1.

Aplique los algoritmos:

- Preferente por amplitud
- Costo uniforme
- Preferente por profundidad

	0	1	2	3
0				
1				
2				
3				

Búsqueda no informada





Búsqueda no informada

- El ratón súper inteligente.

Considere un ratón que aplica algoritmos de búsqueda para desplazarse en el ambiente.

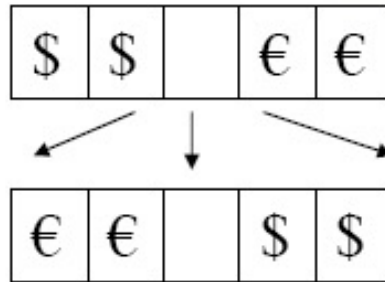
- El costo de cada movimiento es 1.
- El daño causado por el gato cuesta 5 y el de la ratonera 3

Aplique el algoritmo de costo uniforme

	0	1	2	3
0				
1				
2				
3				

Búsqueda no informada

- El problema de Euros y Dólares



Operadores

- *Intercambiar posición de un símbolo con su adyacente (el símbolo adyacente puede ser el espacio en blanco), costo 1*
- *Saltar símbolo (se puede saltar cualquier cantidad de símbolos), costo 2*