

# Matemáticas Discretas II

Oscar Bedoya

`oscar.bedoya@correounivalle.edu.co`

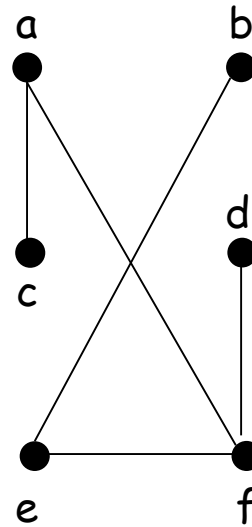
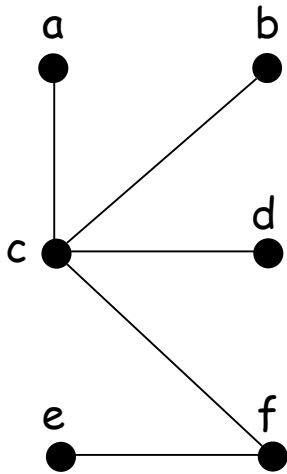
- Árboles
- Aplicaciones de los árboles
- Recorridos de los árboles
- Algoritmo de Prim
- Algoritmo de Kruskal

# Árboles

---

## Árboles

Un árbol es un grafo conexo que no tiene circuitos

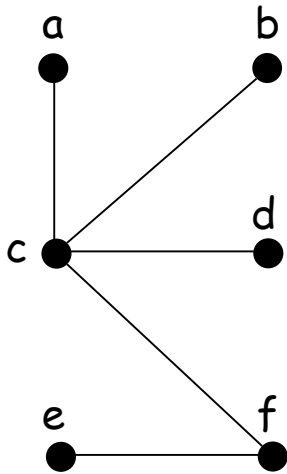


# Árboles

---

## Árboles

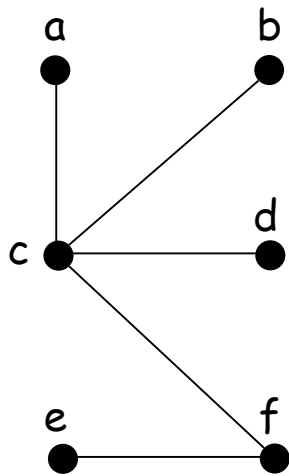
Un árbol es un grafo conexo que no tiene circuitos



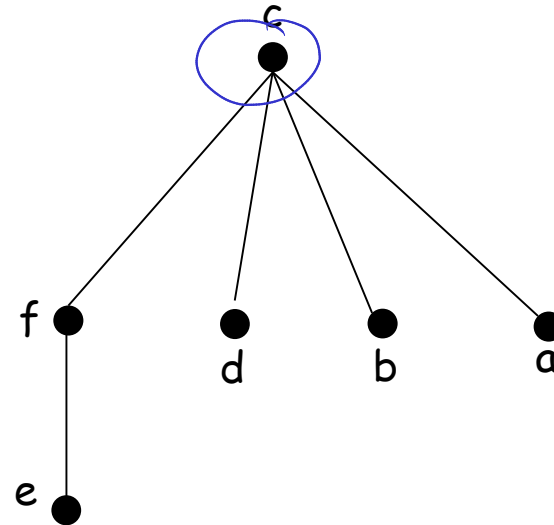
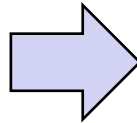
# Árboles

## Árboles

Un árbol es un grafo conexo que no tiene circuitos

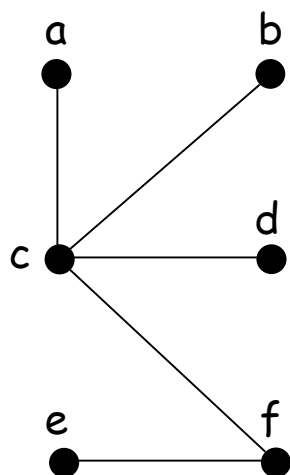


Árbol

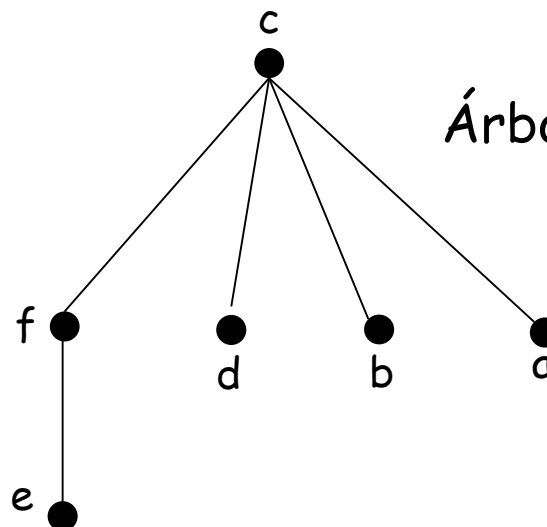
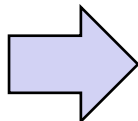


Árbol con raíz en c

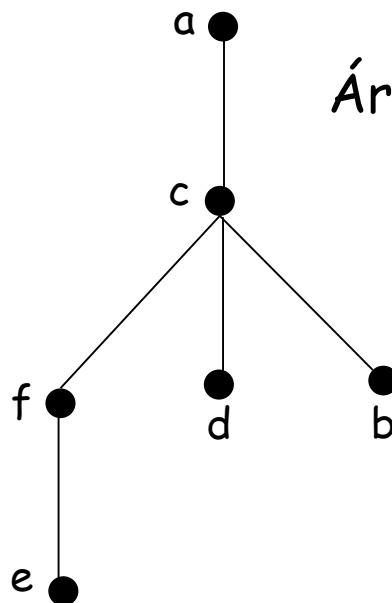
# Árboles



Árbol



Árbol con raíz en c



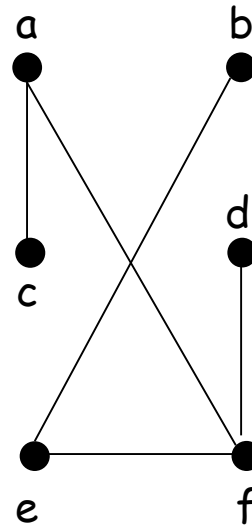
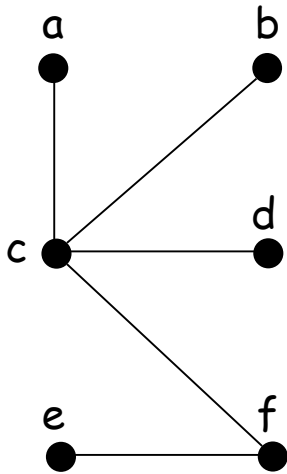
Árbol con raíz en a

# Árboles

---

## Árboles

Un árbol es un grafo conexo que no tiene circuitos

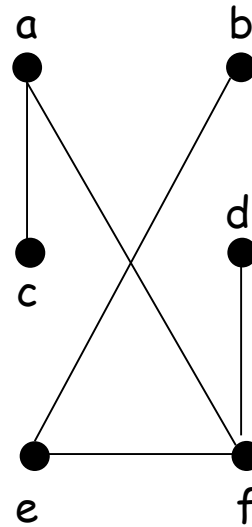


# Árboles

---

## Árboles

Un árbol es un grafo conexo que no tiene circuitos



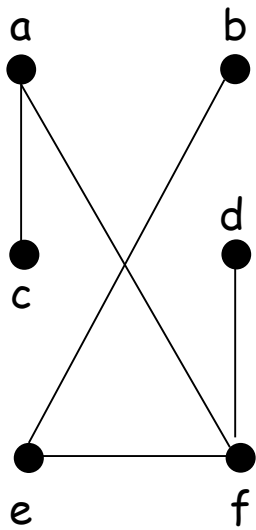


# Árboles

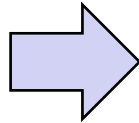
---

## Árboles

Un árbol es un grafo conexo que no tiene circuitos



Árbol

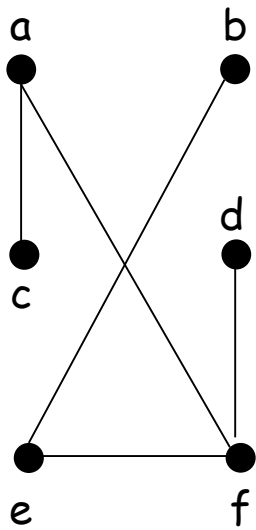


Árbol con raíz en e

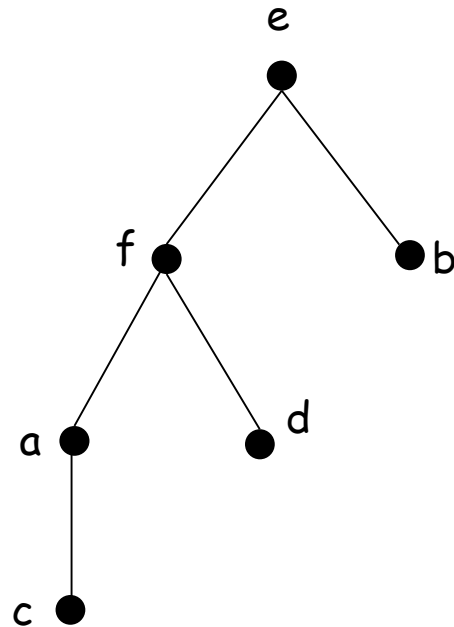
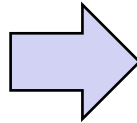
# Árboles

## Árboles

Un árbol es un grafo conexo que no tiene circuitos



Árbol



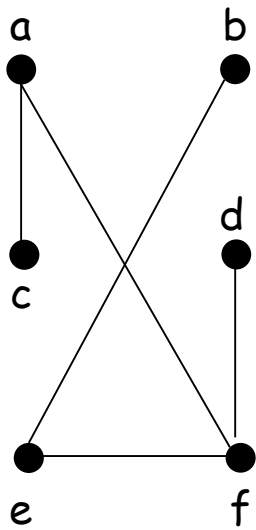
Árbol con raíz en e

# Árboles

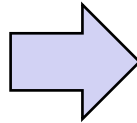
---

## Árboles

Un árbol es un grafo conexo que no tiene circuitos



Árbol

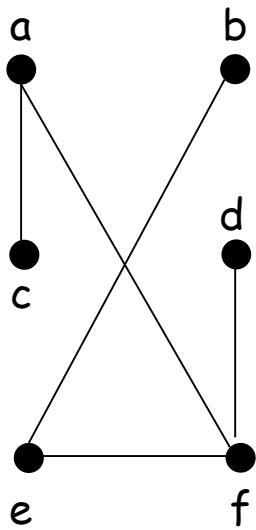


Árbol con raíz en a

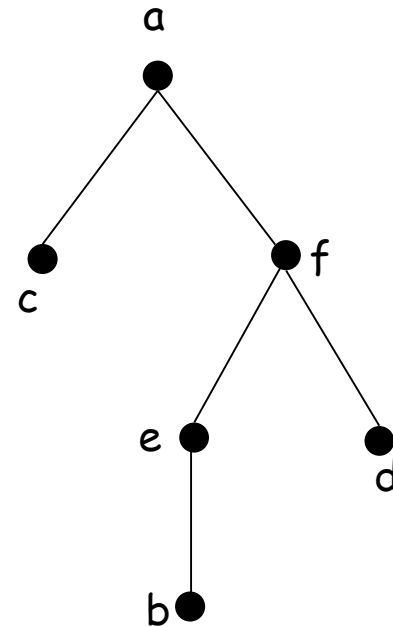
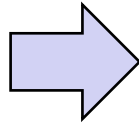
# Árboles

## Árboles

Un árbol es un grafo conexo que no tiene circuitos



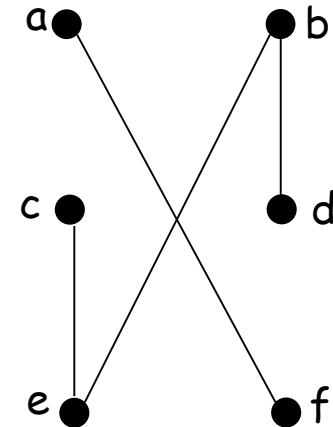
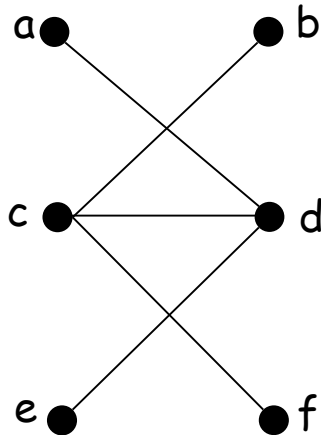
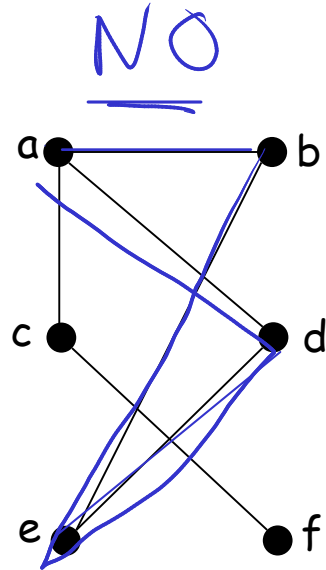
Árbol



Árbol con raíz en a

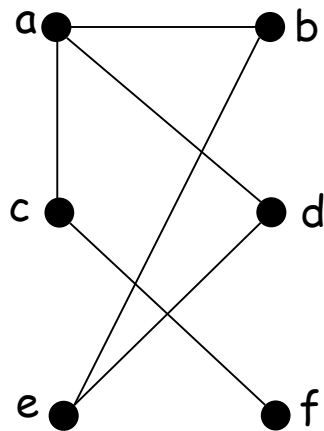
# Árboles

Indicar cuáles de los siguientes grafos son árboles

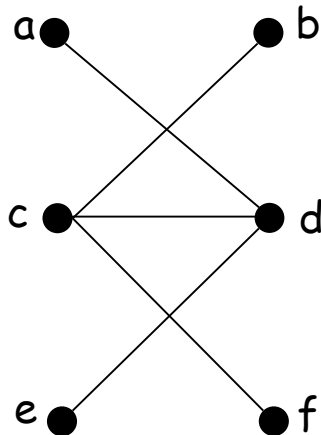


# Árboles

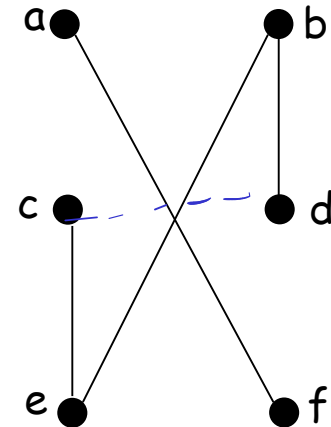
Indicar cuáles de los siguientes grafos son árboles



No es árbol porque  
hay un circuito  
( $a \rightarrow d \rightarrow e \rightarrow b \rightarrow a$ )



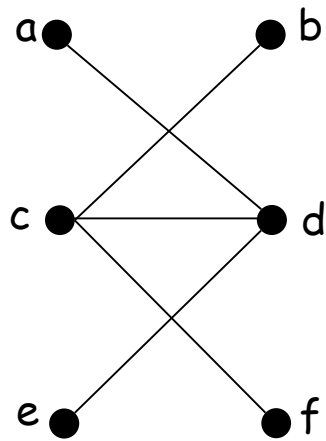
Es un árbol



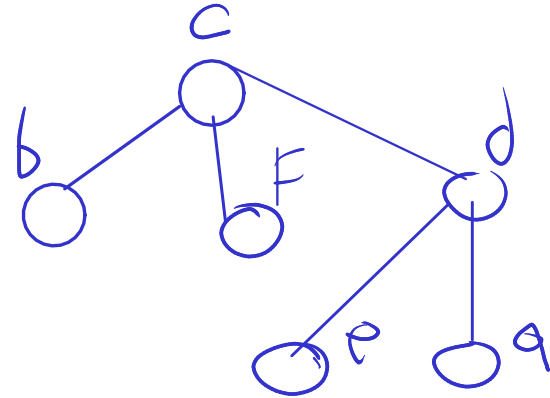
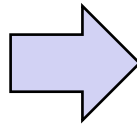
No es árbol porque  
no es conexo

# Árboles

---



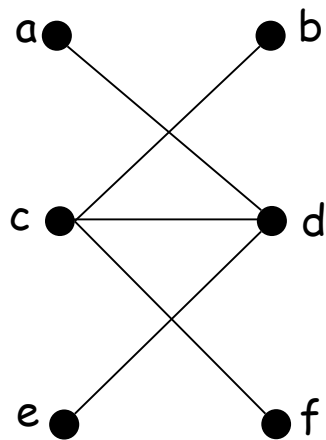
Es un árbol



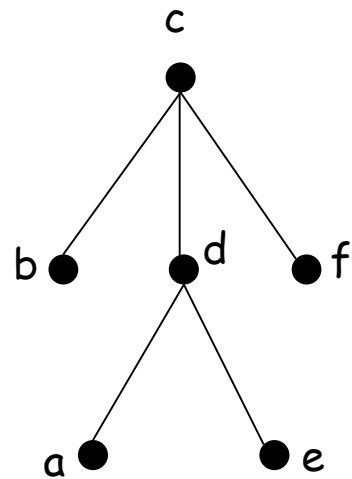
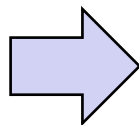
Árbol con raíz en c

# Árboles

---



Es un árbol

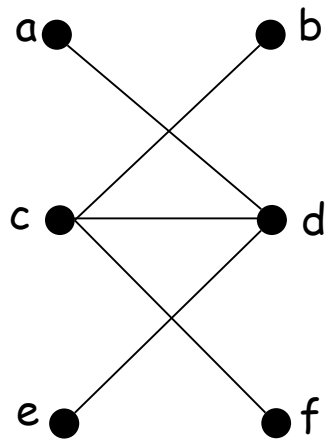


Árbol con raíz en c

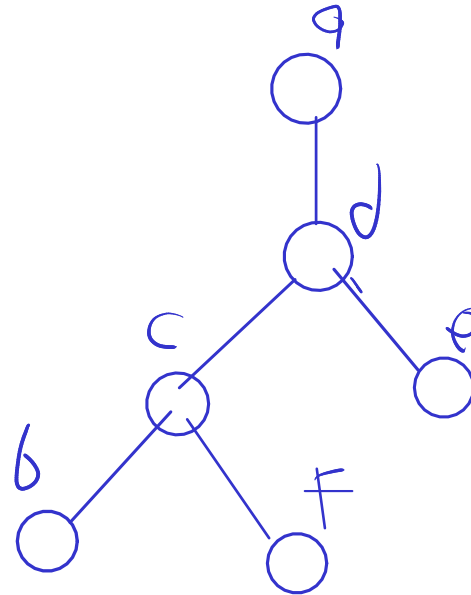
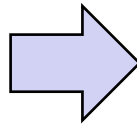


# Árboles

---



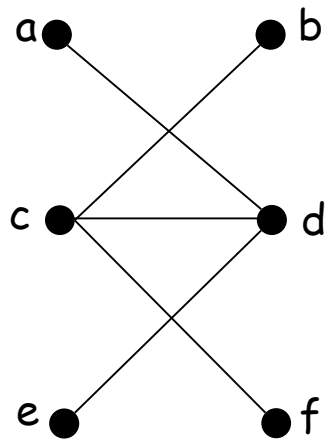
Es un árbol



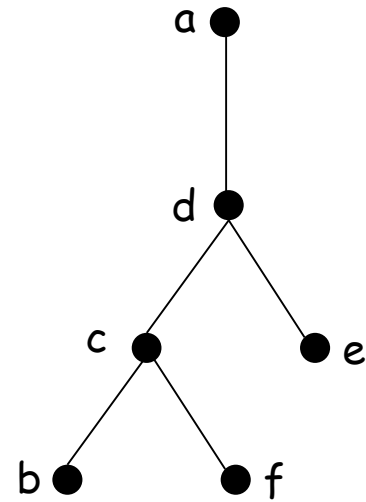
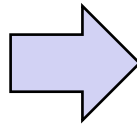
Árbol con raíz en a

# Árboles

---



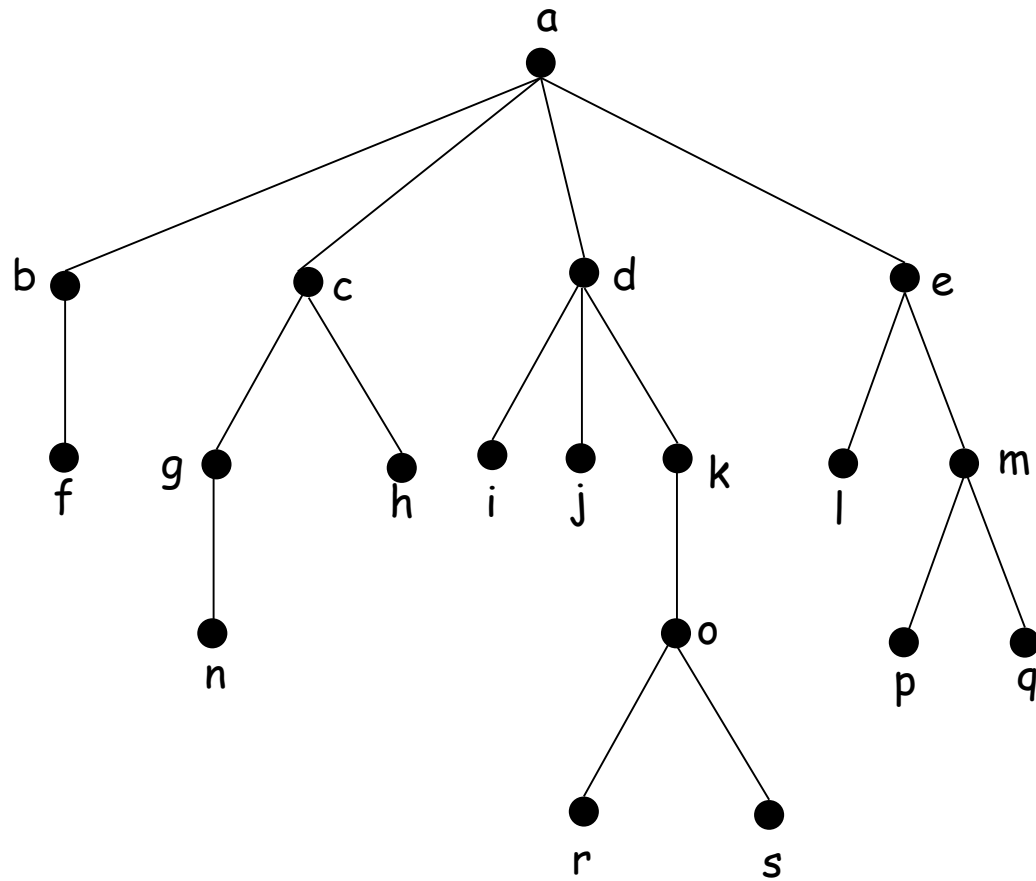
Es un árbol



Árbol con raíz en a

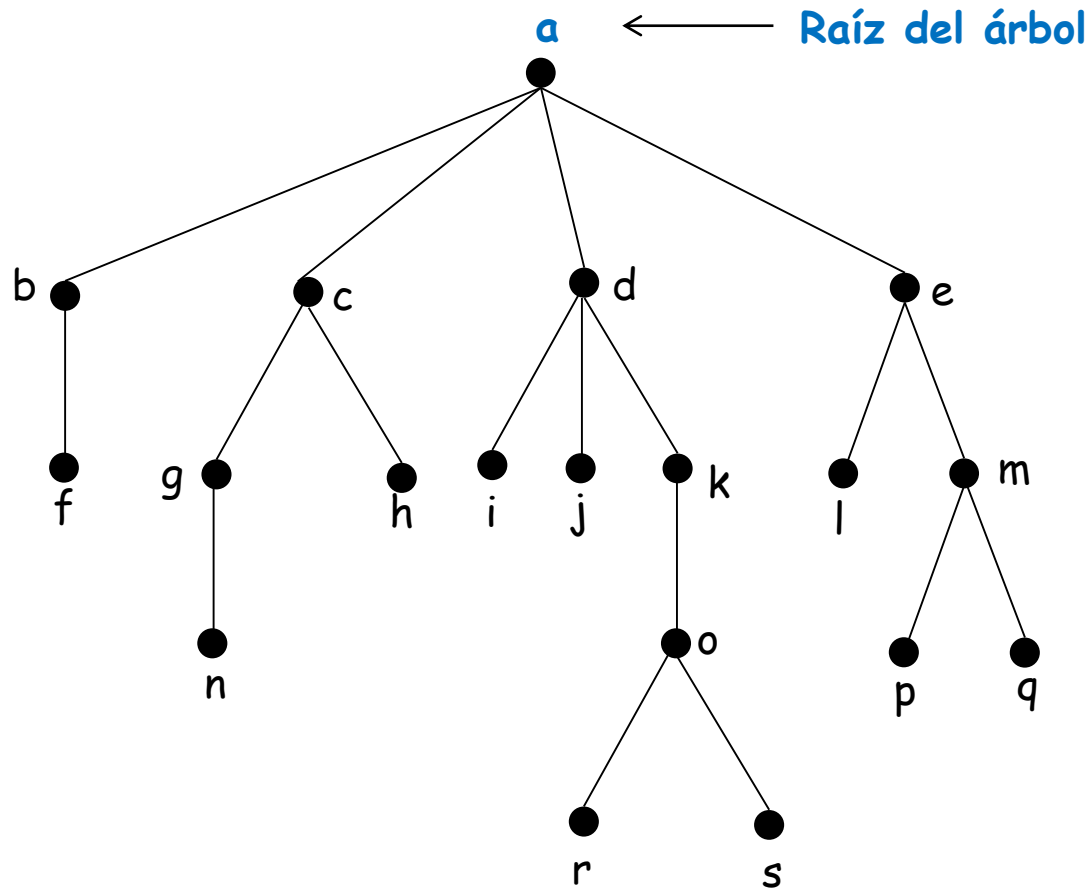
# Árboles

---



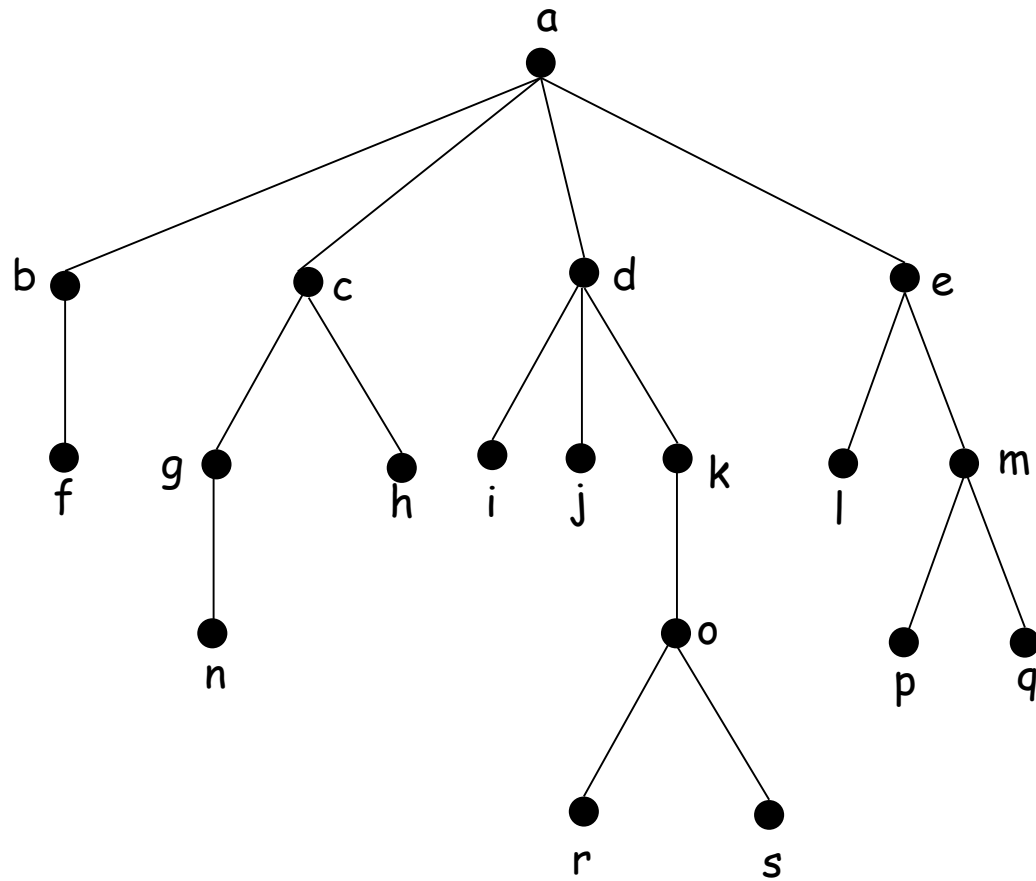
# Árboles

---



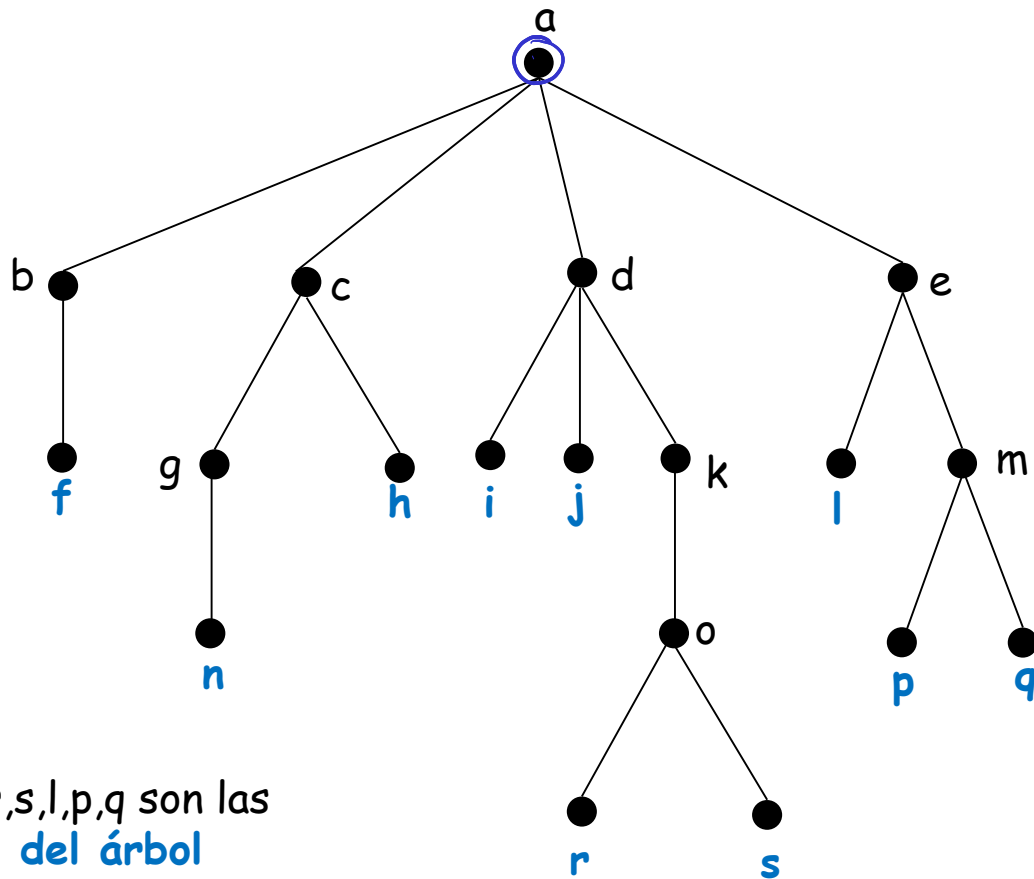
# Árboles

---



# Árboles

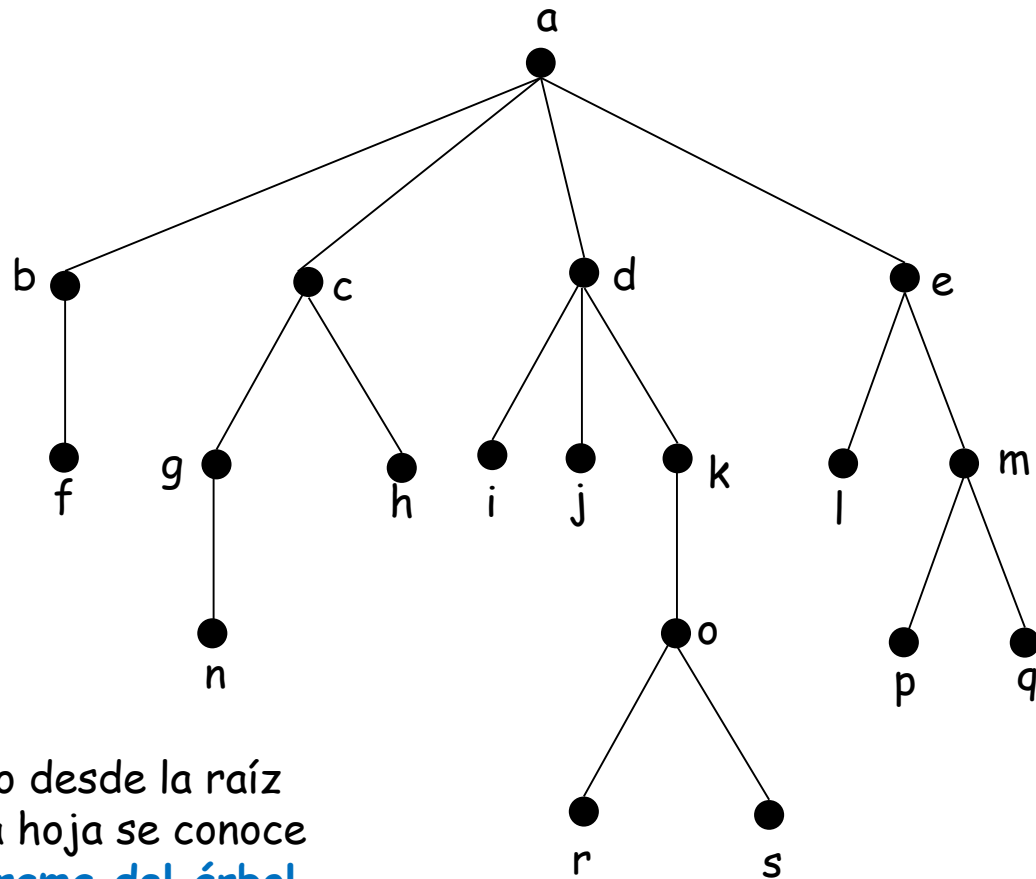
---



f,n,h,i,j,r,s,l,p,q son las  
hojas del árbol

# Árboles

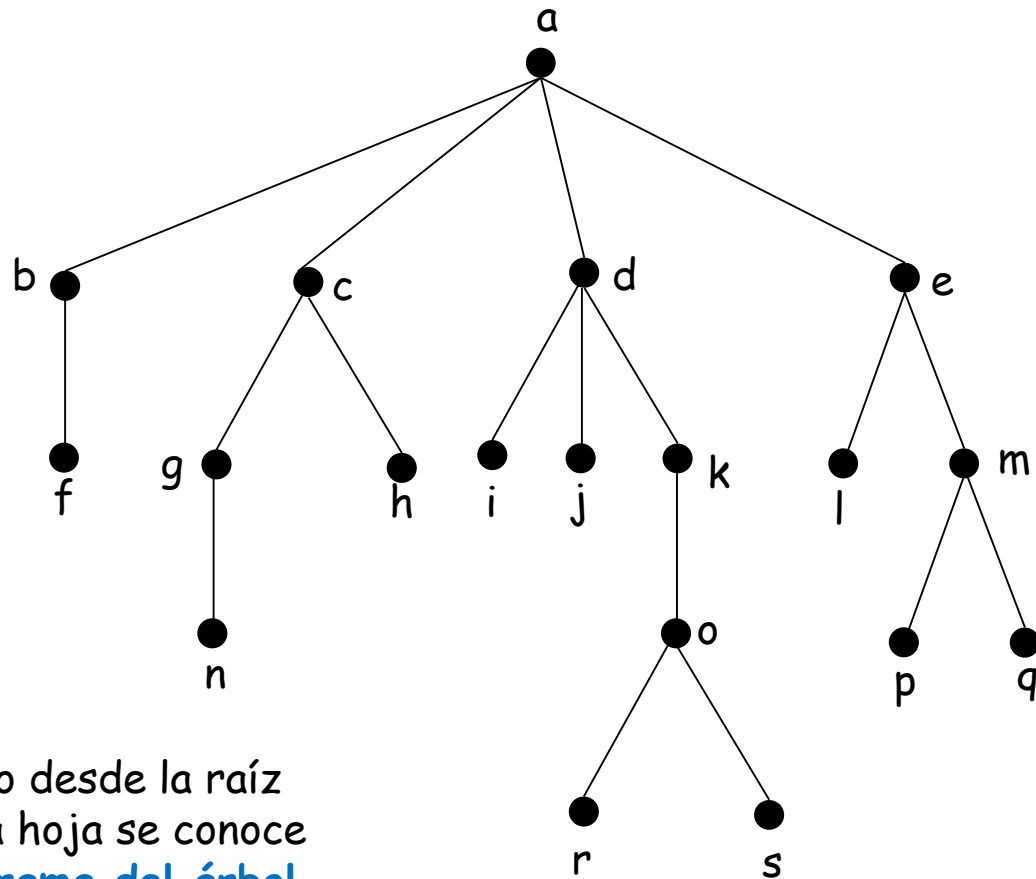
---



Un camino desde la raíz hasta cada hoja se conoce como una **rama del árbol**

# Árboles

---

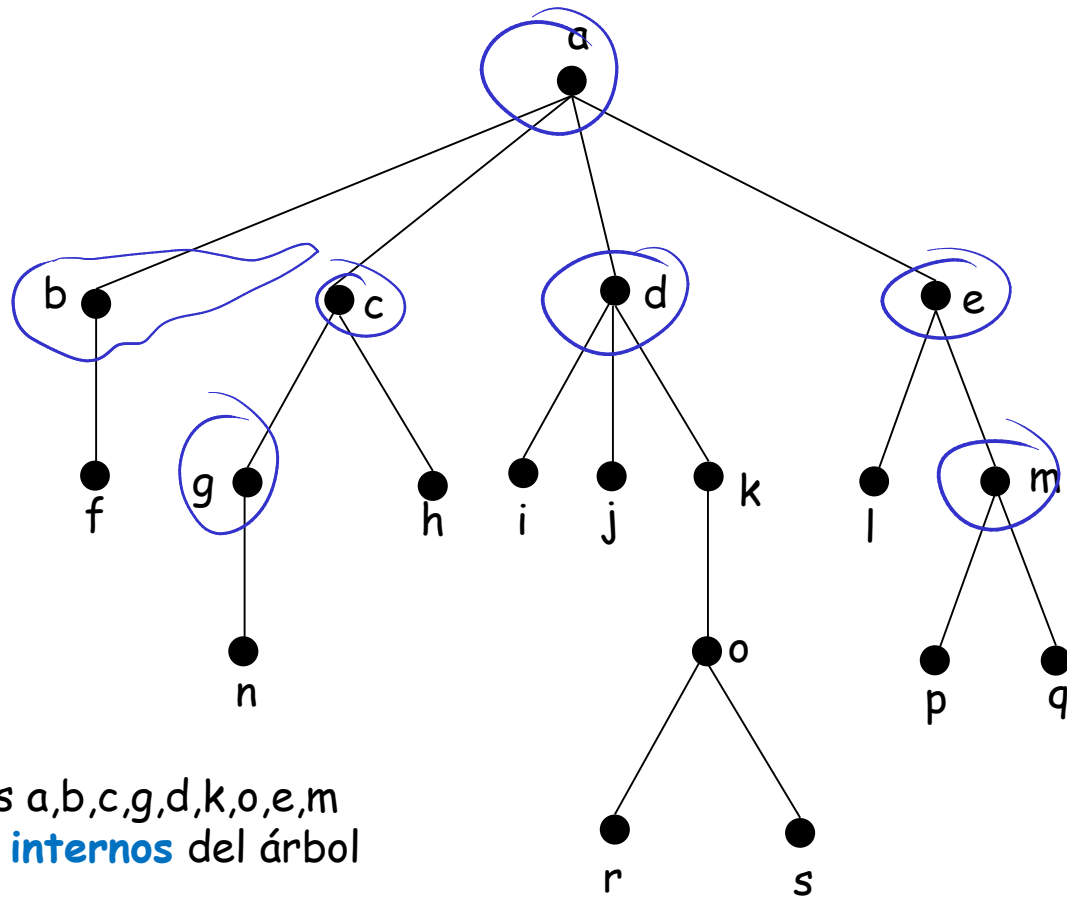


Un camino desde la raíz  
hasta cada hoja se conoce  
como una **rama del árbol**

$a \rightarrow b \rightarrow f$   
 $a \rightarrow d \rightarrow k \rightarrow o \rightarrow s$



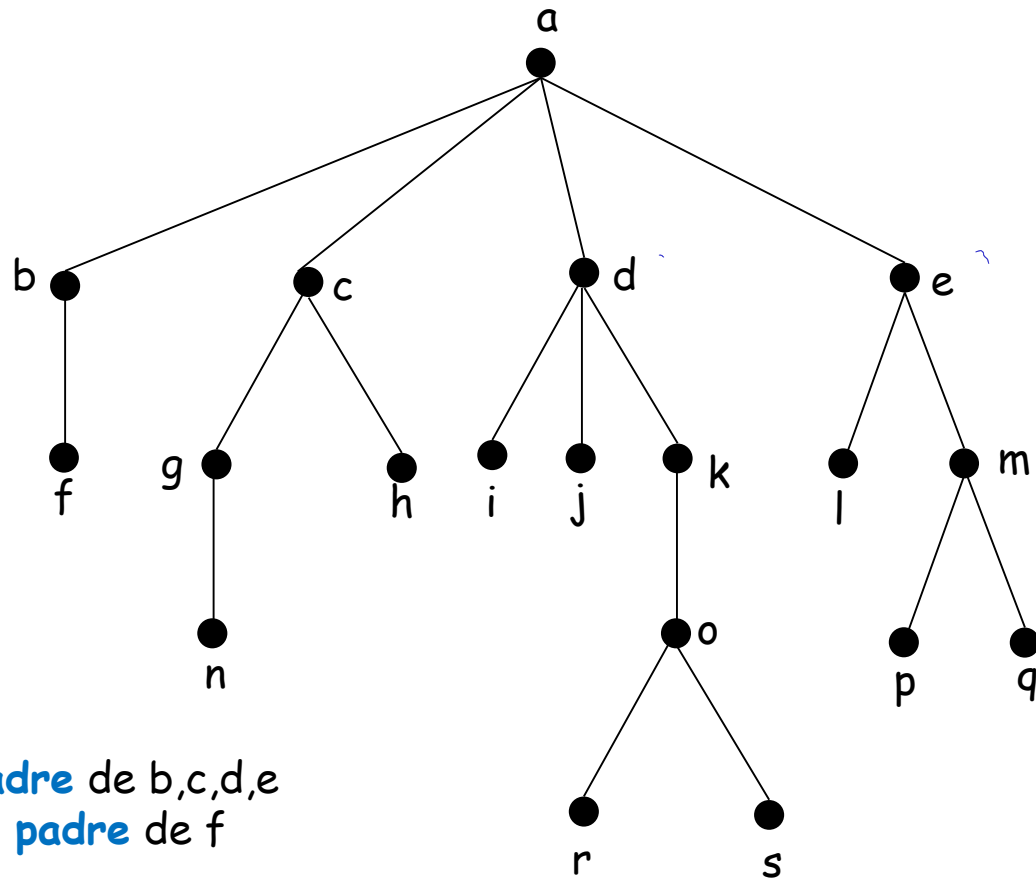
# Árboles



Los nodos a,b,c,g,d,k,o,e,m  
son **nodos internos** del árbol

# Árboles

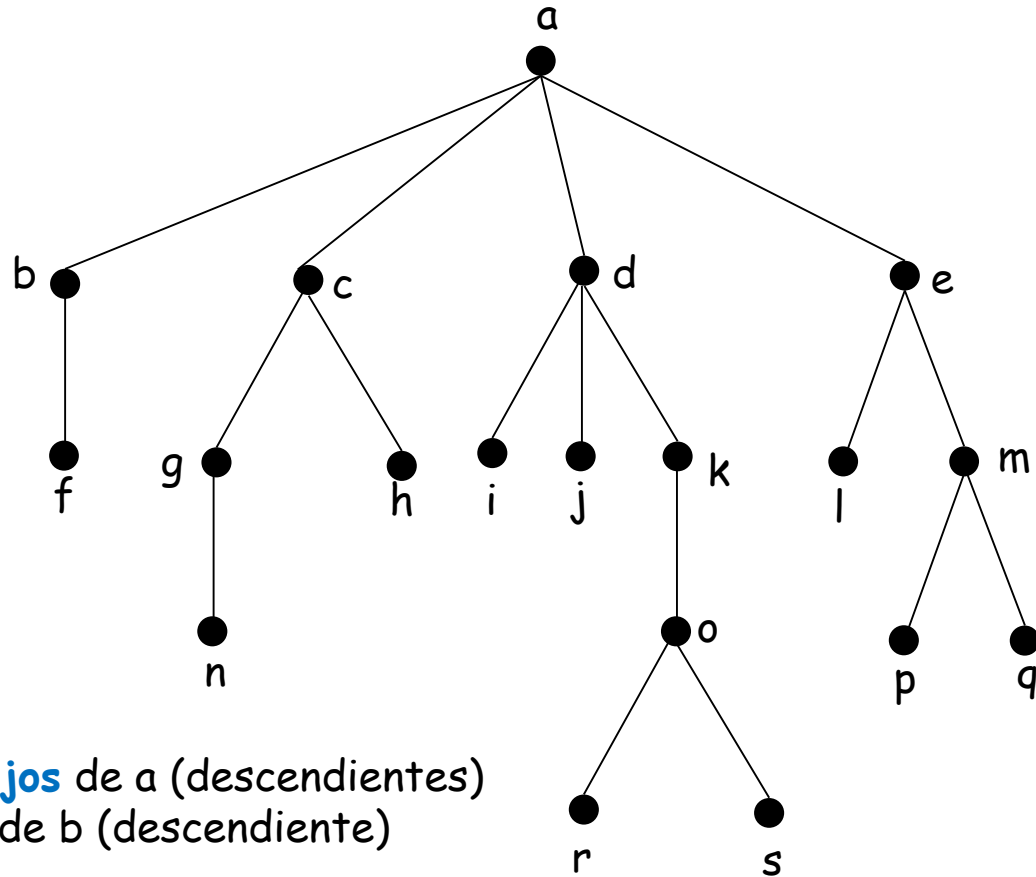
---



a es el **padre** de b,c,d,e  
b es el **padre** de f

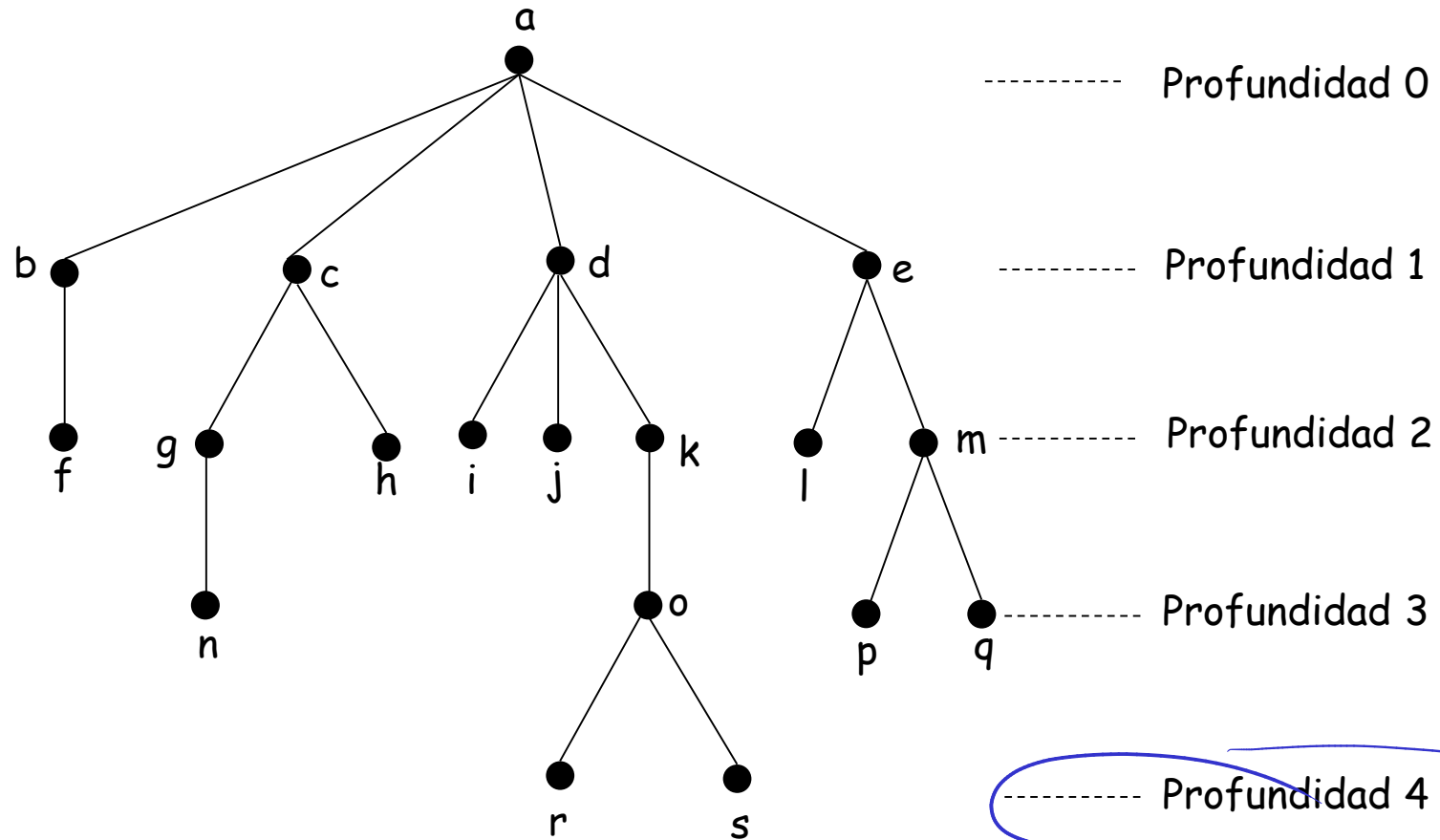
# Árboles

---

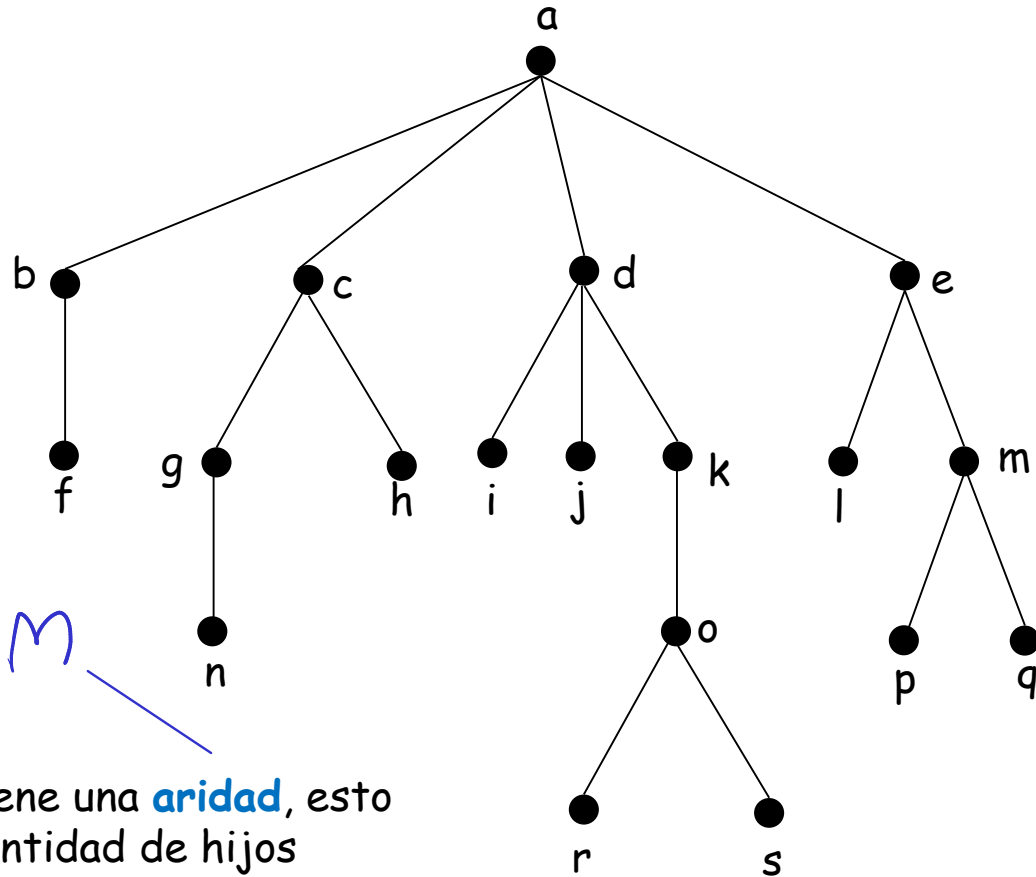


b,c,d,e son los **hijos** de a (descendientes)  
f es el **hijo** de b (descendiente)

# Árboles



# Árboles

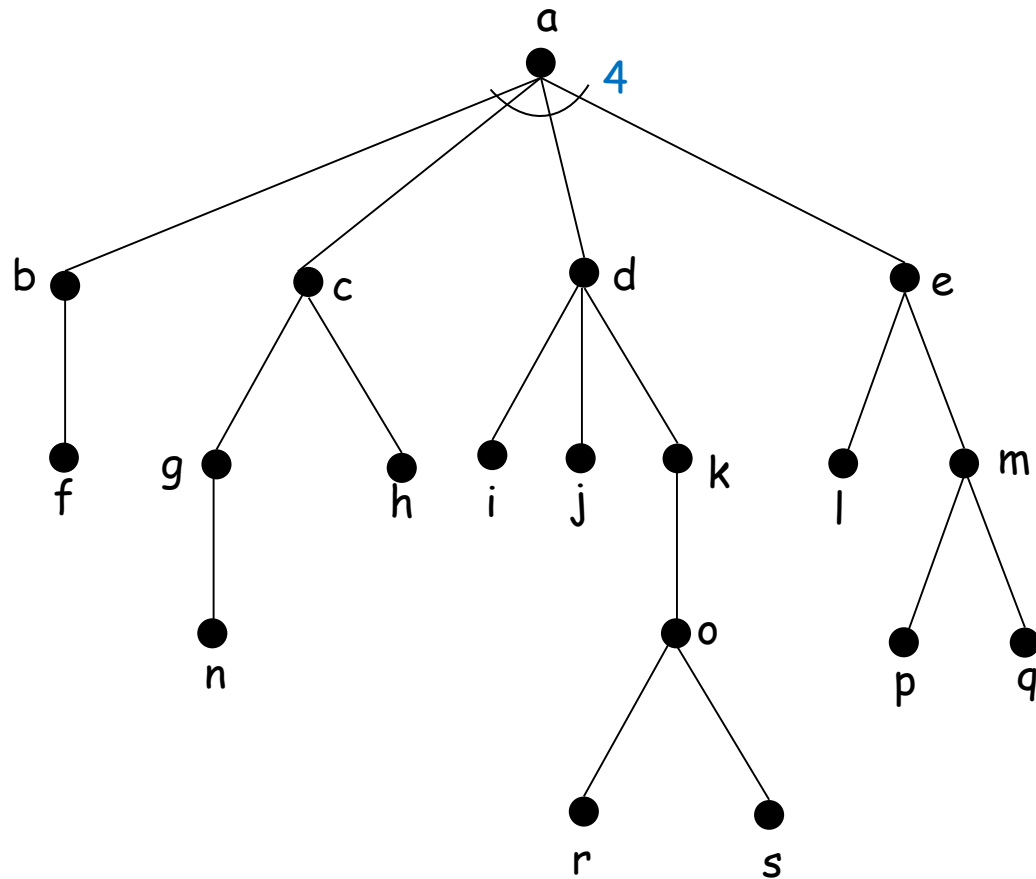


M

Cada nodo tiene una **aridad**, esto es, la cantidad de hijos

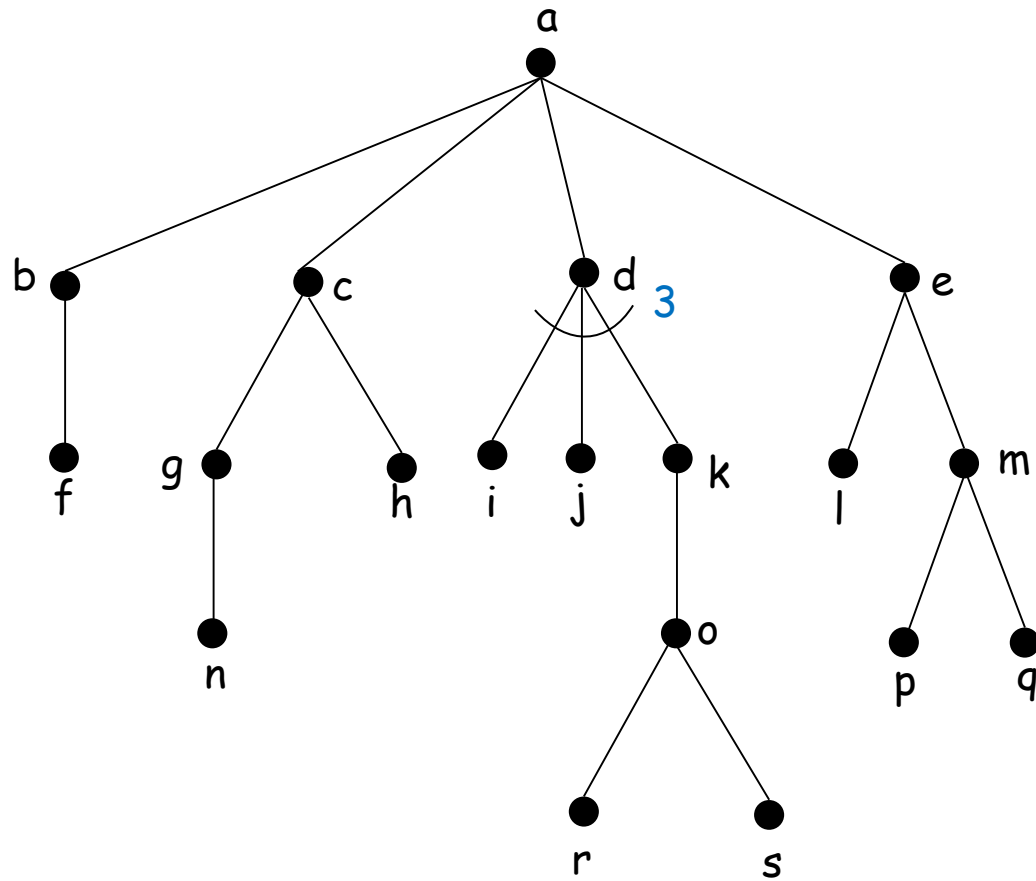
# Árboles

---



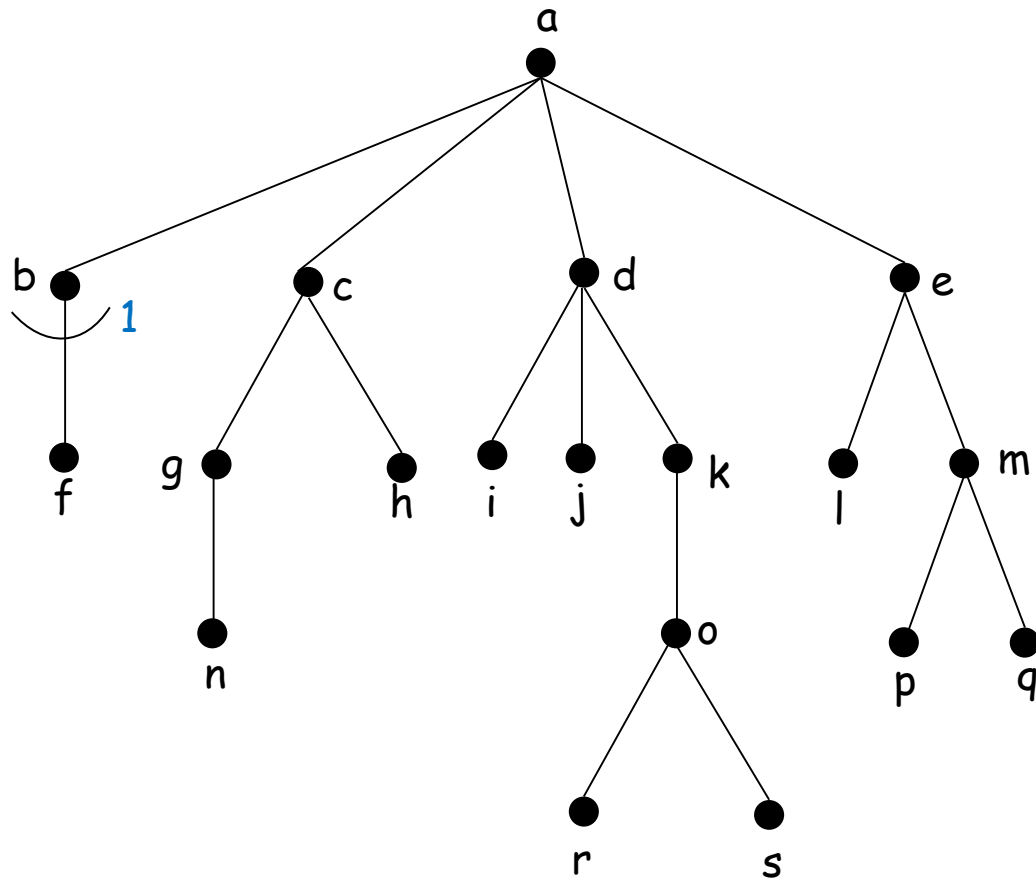
# Árboles

---



# Árboles

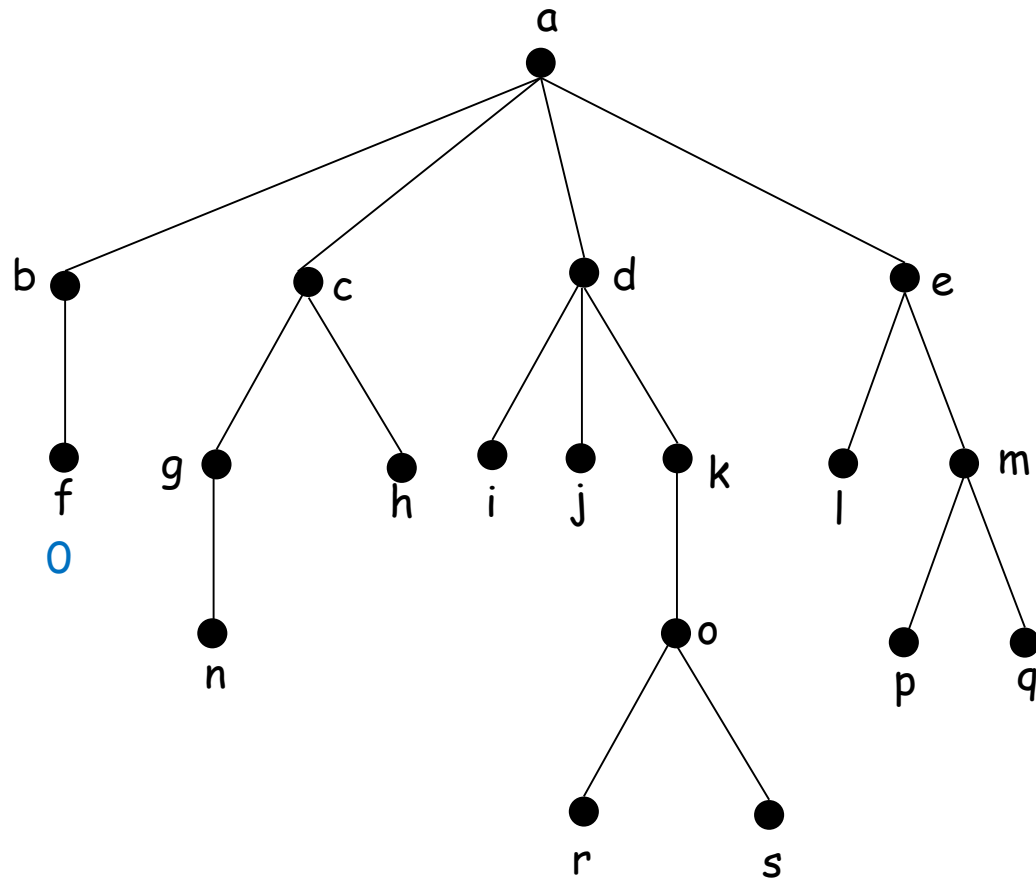
---





# Árboles

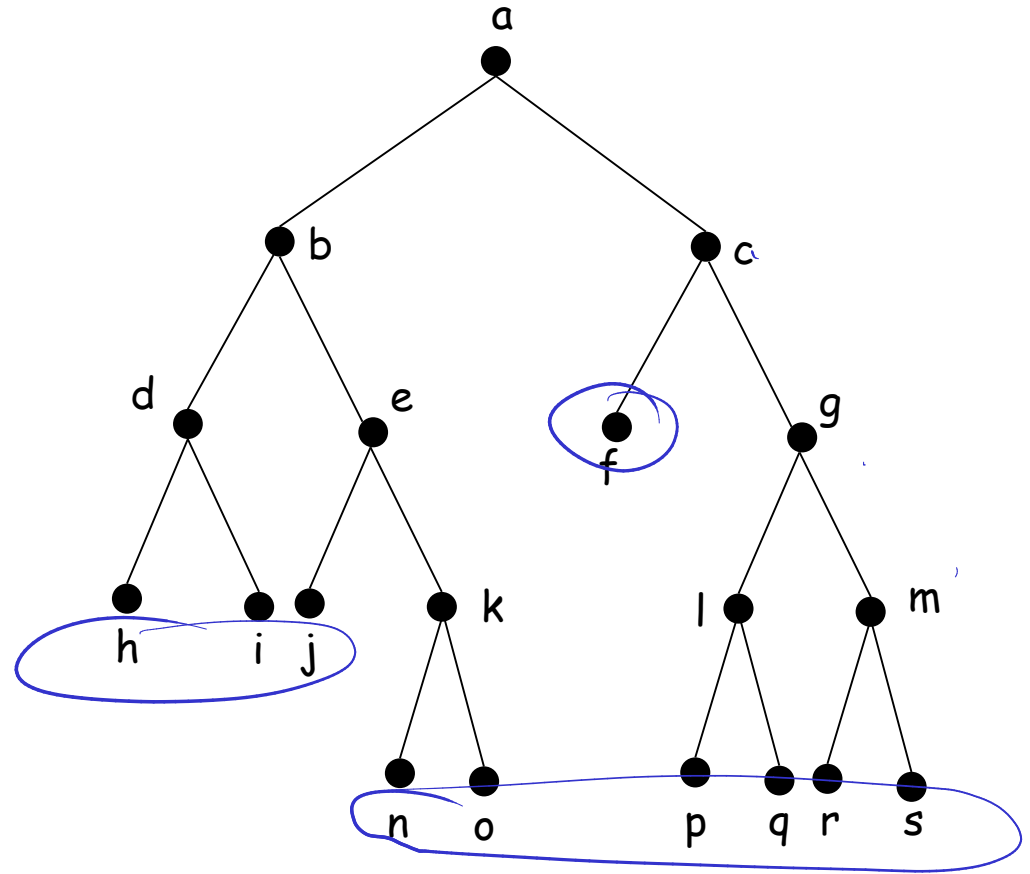
---



# Árboles

Dado el siguiente árbol indique:

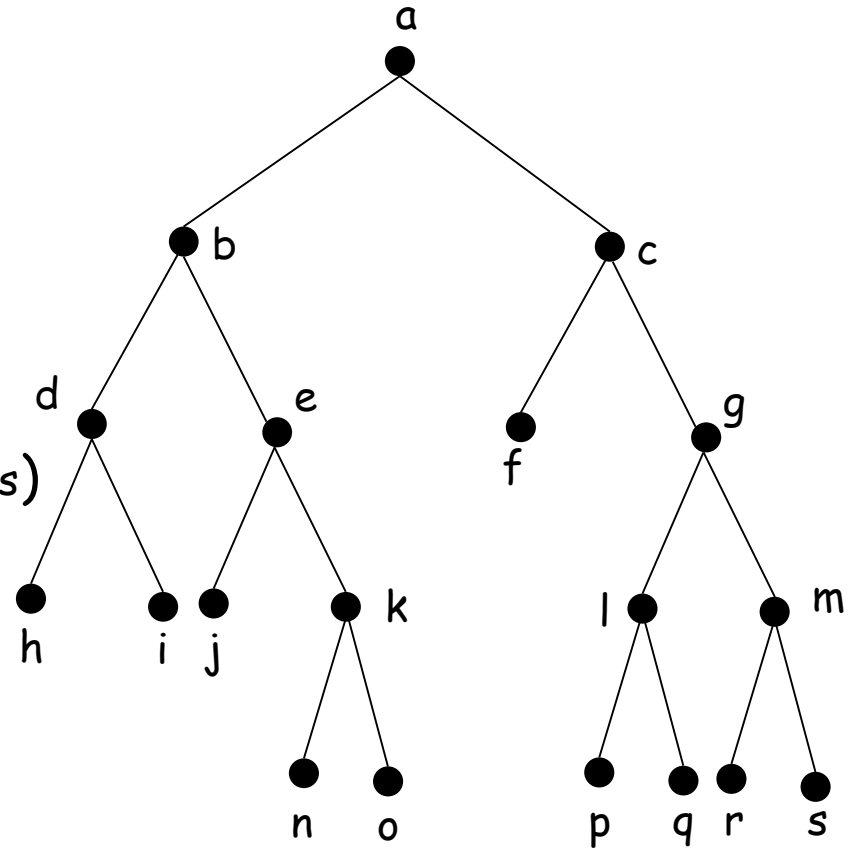
- La raíz del árbol  $a$
- El padre de e  $b$
- Los hijos de g  $l, m$
- Las hojas del árbol
- La profundidad de m  $3$
- La profundidad de b  $1$
- La aridad de c  $2$
- La aridad de j  $0$



# Árboles

Dado el siguiente árbol indique:

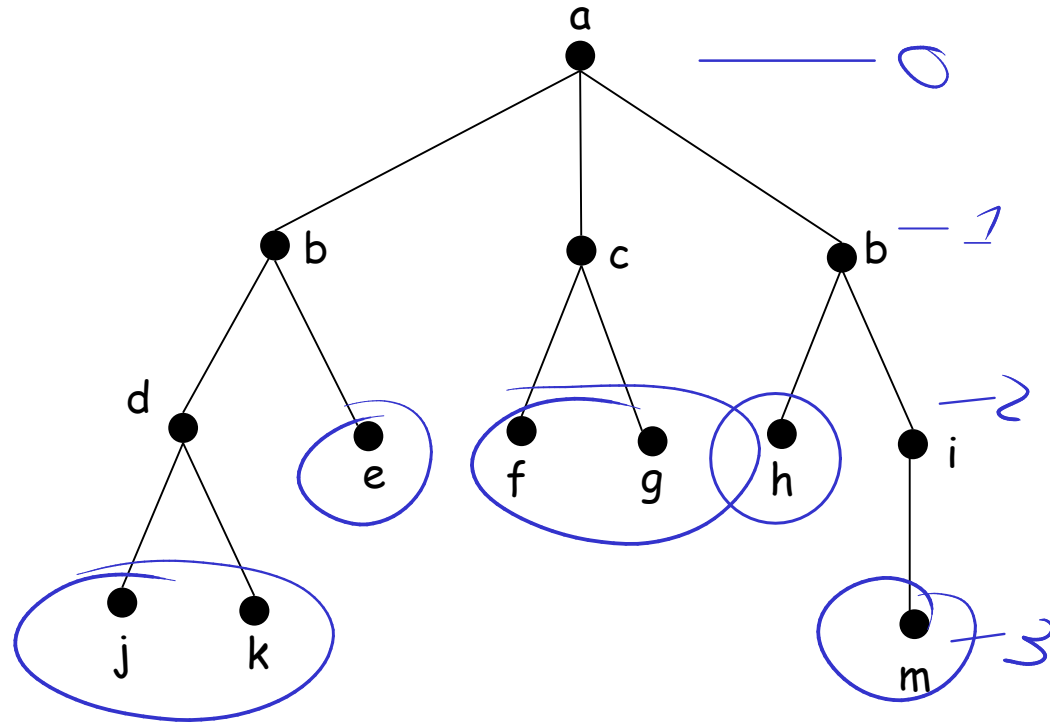
- La raíz del árbol (a)
- El padre de e (b)
- Los hijos de g (l,m)
- Las hojas del árbol (h,i,j,n,o,f,p,q,r,s)
- La profundidad de m (3)
- La profundidad de b (1)
- La aridad de c (2)
- La aridad de j (0)



# Árboles

Dado el siguiente árbol indique:

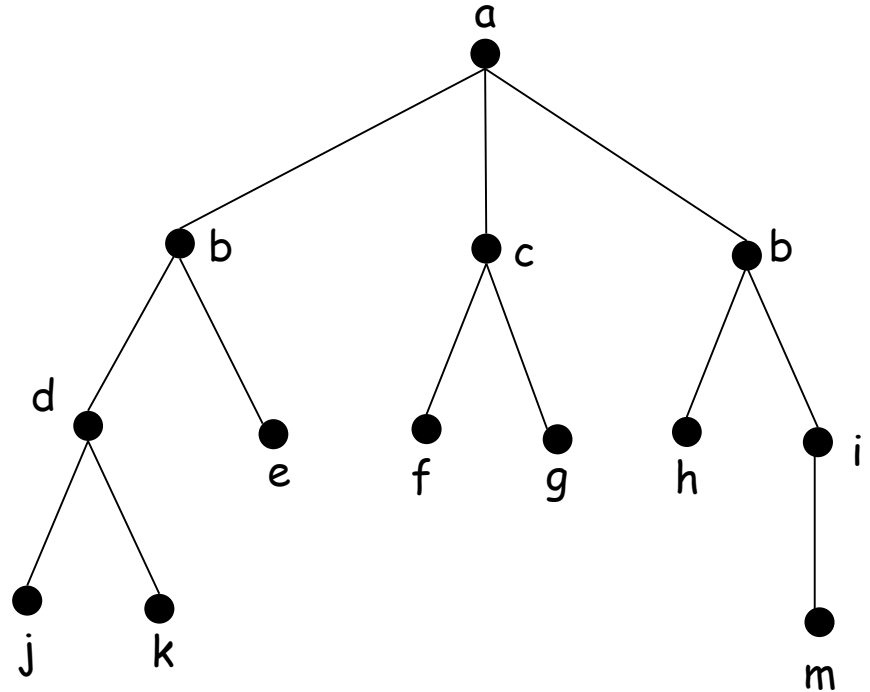
- La raíz del árbol **a**
- El padre de i **b**
- Los hijos de d **j k**
- Las hojas del árbol
- La profundidad de e **2**
- La profundidad de a **0**
- La aridad de i **1**
- La aridad de a **3**



# Árboles

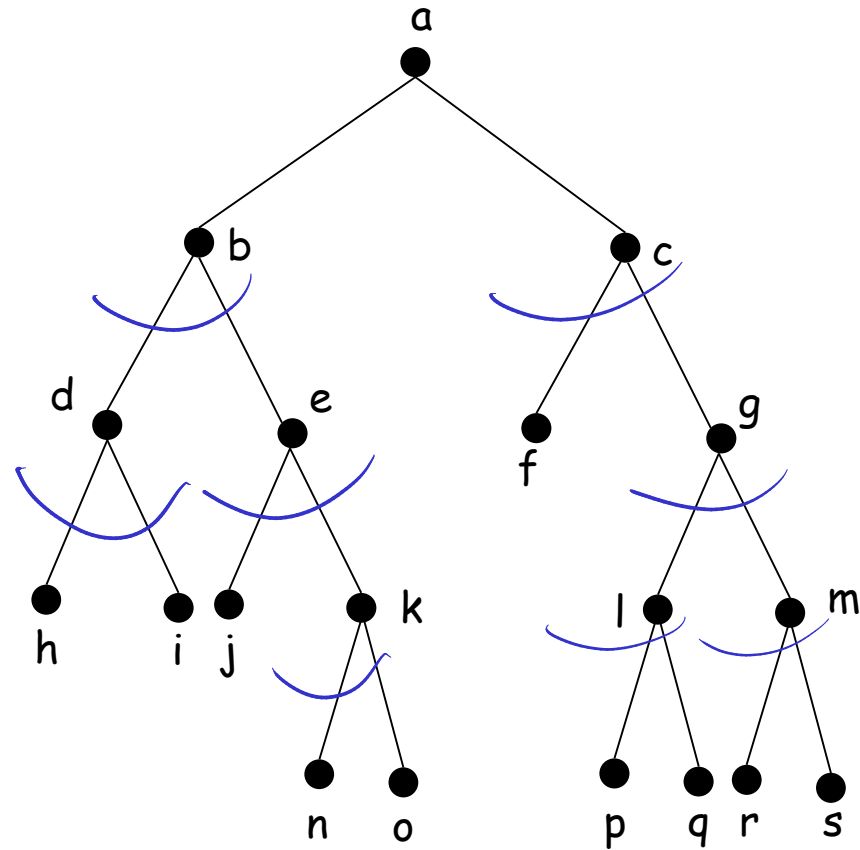
Dado el siguiente árbol indique:

- La raíz del árbol (a)
- El padre de i (b)
- Los hijos de d (j,k)
- Las hojas del árbol (j,k,e,f,g,h,m)
- La profundidad de e (2)
- La profundidad de a (0)
- La aridad de i (1)
- La aridad de a (3)



# Árboles

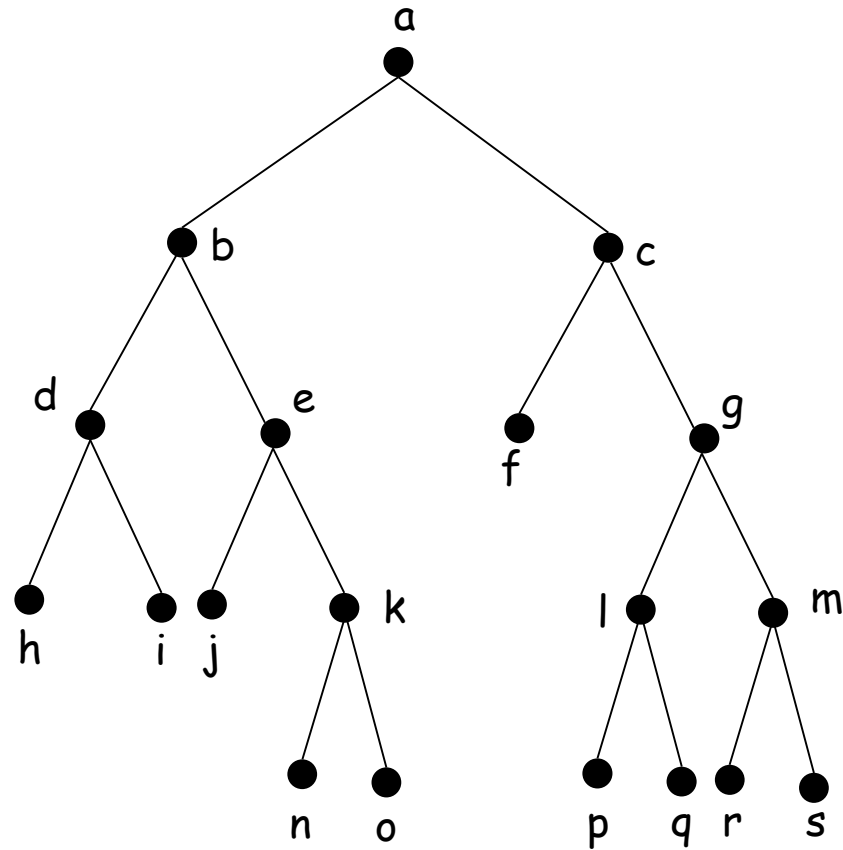
---



Cuál es la aridad de los nodos internos

# Árboles

---



Todos los nodos internos tienen la misma aridad

```

graph TD
    Root(( )) --- L1_1(( ))
    Root --- L1_2(( ))
    Root --- L1_3(( ))
    L1_1 --- L2_1(( ))
    L1_1 --- L2_2(( ))
    L1_1 --- L2_3(( ))
    L2_1 --- L3_1(( ))
    L2_1 --- L3_2(( ))
    L2_1 --- L3_3(( ))
    L2_2 --- L3_4(( ))
    L2_2 --- L3_5(( ))
    L2_2 --- L3_6(( ))
    L3_5 --- L4_1(( ))
    L3_5 --- L4_2(( ))
    L3_5 --- L4_3(( ))
    L1_2 --- L2_4(( ))
    L1_2 --- L2_5(( ))
    L1_2 --- L2_6(( ))
    L2_4 --- L3_7(( ))
    L2_4 --- L3_8(( ))
    L2_4 --- L3_9(( ))
    L2_5 --- L3_10(( ))
    L2_5 --- L3_11(( ))
    L2_5 --- L3_12(( ))
    L2_6 --- L3_13(( ))
    L2_6 --- L3_14(( ))
    L2_6 --- L3_15(( ))
    L3_13 --- L4_4(( ))
    L3_13 --- L4_5(( ))
    L3_13 --- L4_6(( ))
    L3_14 --- L4_7(( ))
    L3_14 --- L4_8(( ))
    L3_14 --- L4_9(( ))
    L3_15 --- L4_10(( ))
    L3_15 --- L4_11(( ))
    L3_15 --- L4_12(( ))
    L1_3 --- L2_16(( ))
    L1_3 --- L2_17(( ))
    L1_3 --- L2_18(( ))
    L2_16 --- L3_16(( ))
    L2_16 --- L3_17(( ))
    L2_16 --- L3_18(( ))
    L2_17 --- L3_19(( ))
    L2_17 --- L3_20(( ))
    L2_17 --- L3_21(( ))
    L2_18 --- L3_22(( ))
    L2_18 --- L3_23(( ))
    L2_18 --- L3_24(( ))
    L3_22 --- L4_13(( ))
    L3_22 --- L4_14(( ))
    L3_22 --- L4_15(( ))
    L3_23 --- L4_16(( ))
    L3_23 --- L4_17(( ))
    L3_23 --- L4_18(( ))
    L3_24 --- L4_19(( ))
    L3_24 --- L4_20(( ))
    L3_24 --- L4_21(( ))
  
```





# Árboles

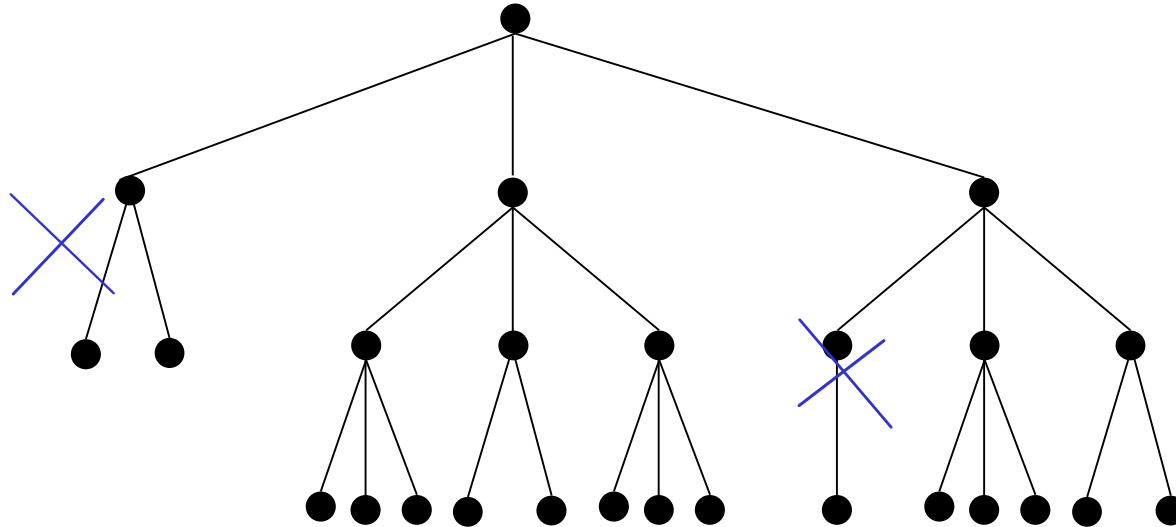
---

## Árbol $n$ -ario completo

Un árbol es  $n$ -ario completo si cada nodo interno tiene exactamente  $n$  hijos

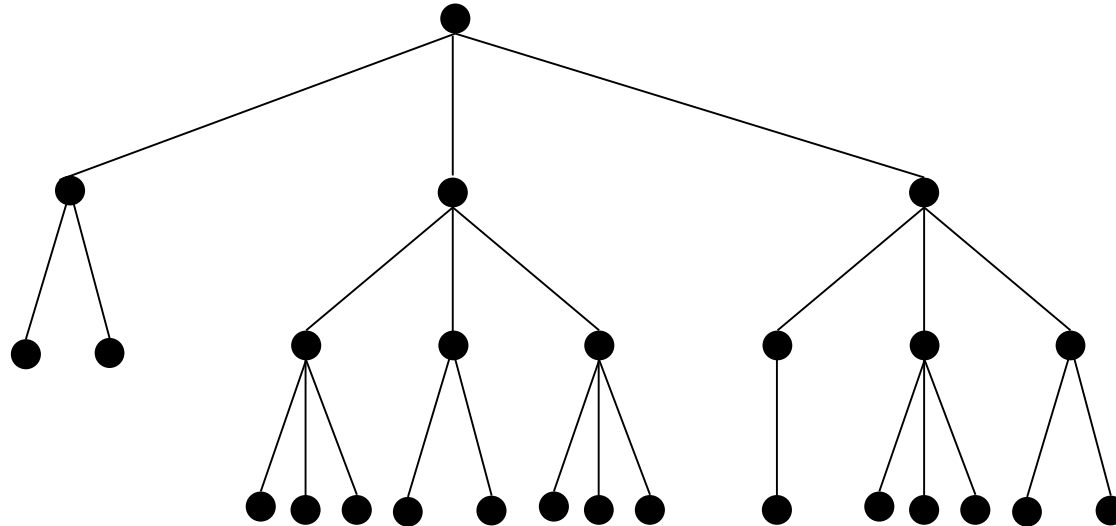
# Árboles

---



# Árboles

---



Cada nodo interno tiene aridad 1, 2, o 3

# Árboles

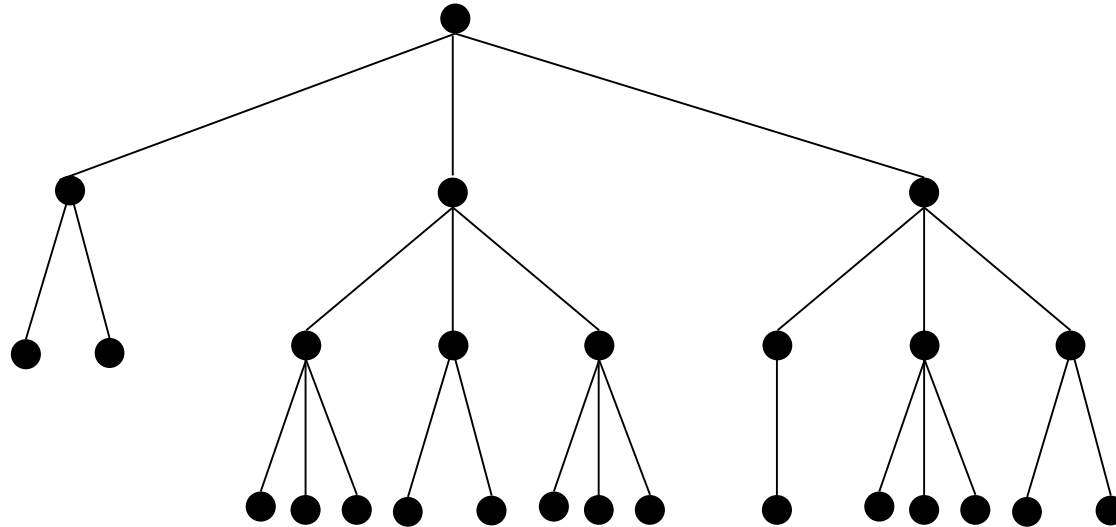
---

## Árbol $n$ -ario

Un árbol es  $n$ -ario si cada nodo interno no tiene más de  $n$  hijos

# Árboles

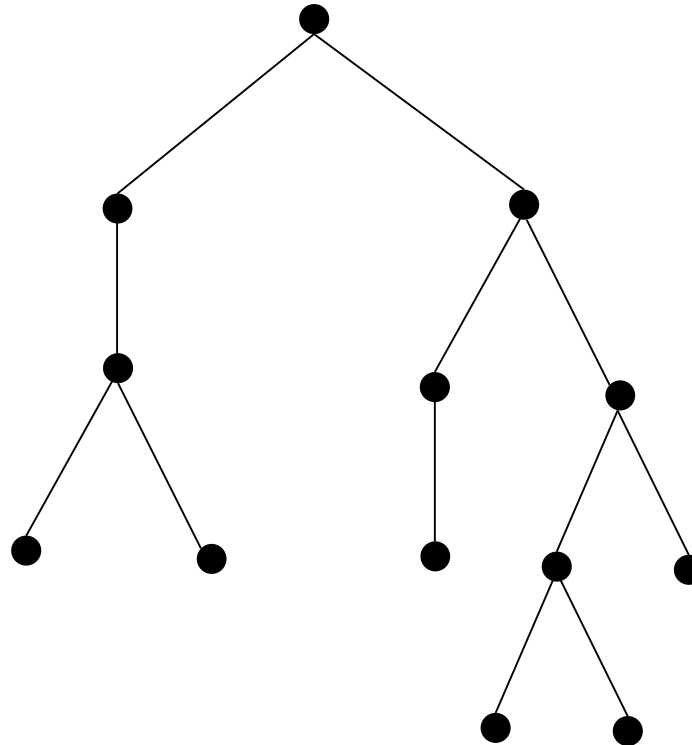
---



Árbol 3-ario

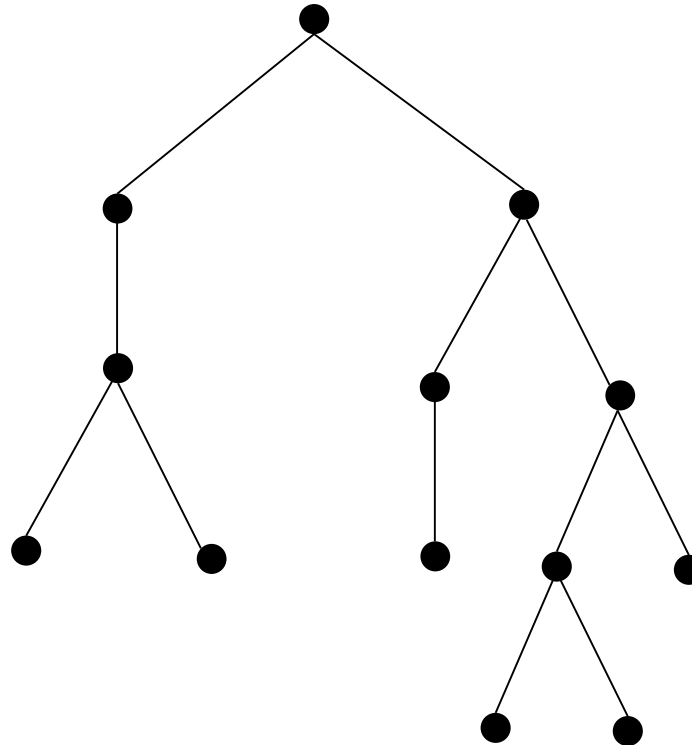
# Árboles

---



# Árboles

---

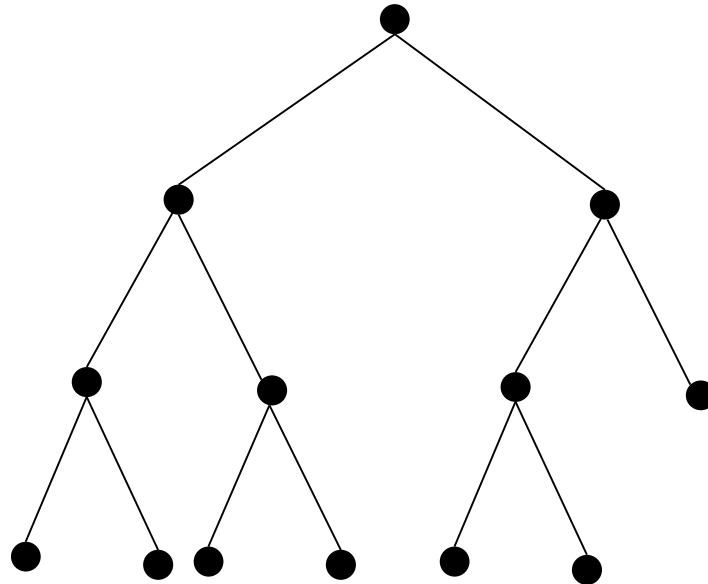


Árbol 2-ario o binario



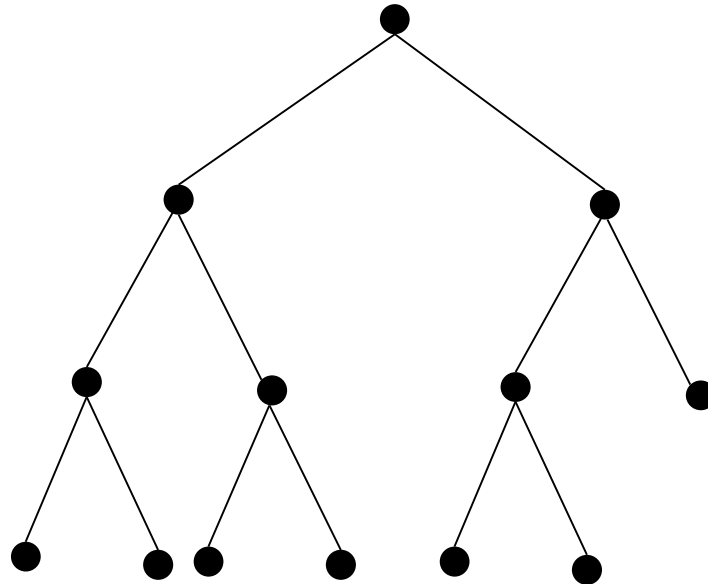
# Árboles

---



# Árboles

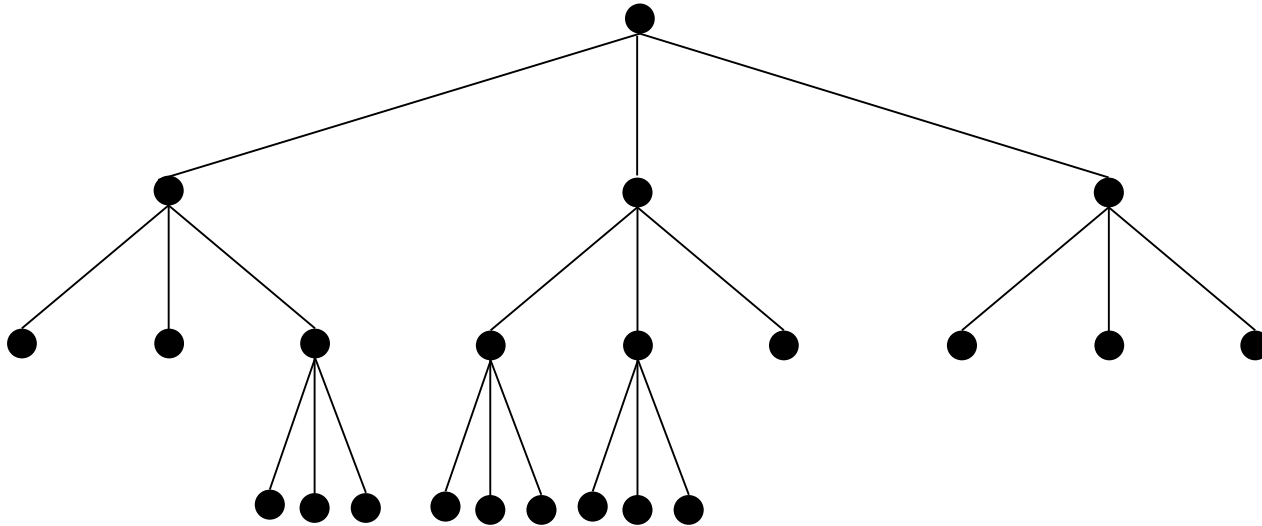
---



Árbol binario completo

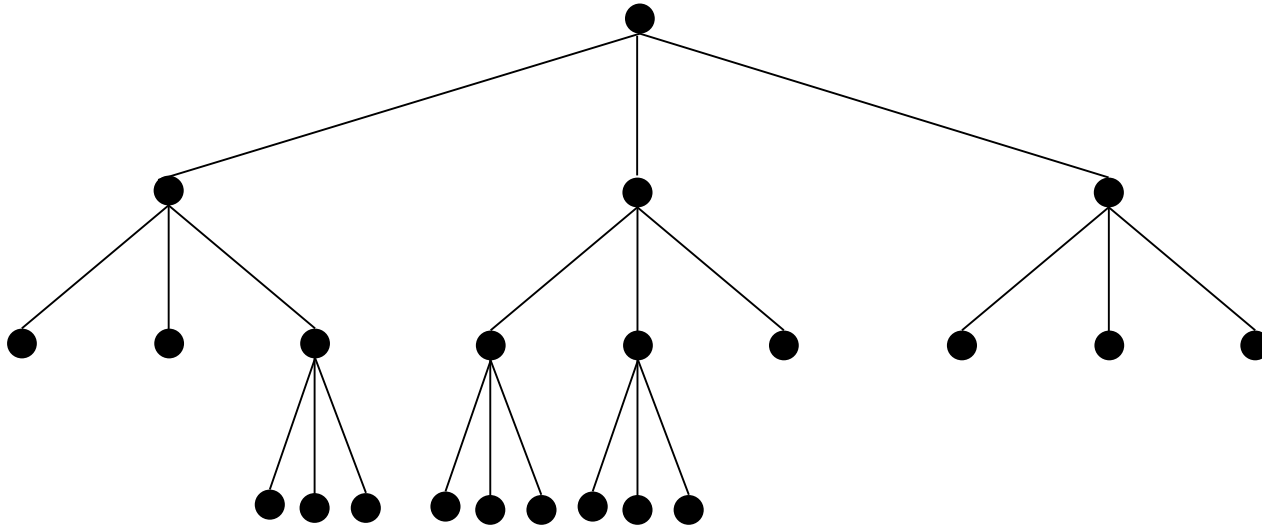
# Árboles

---



# Árboles

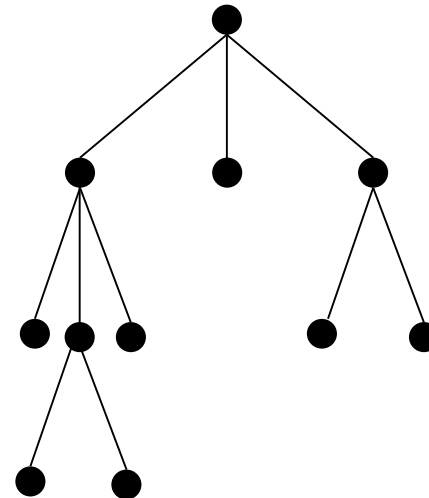
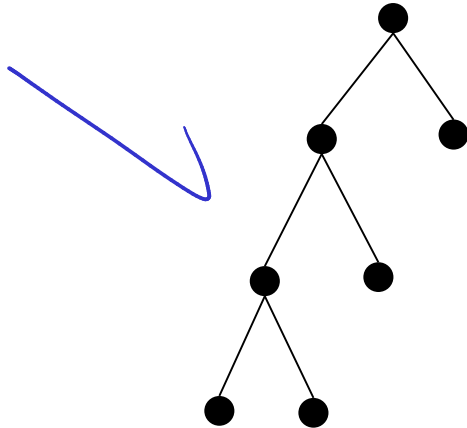
---



Árbol 3-ario completo

# Árboles

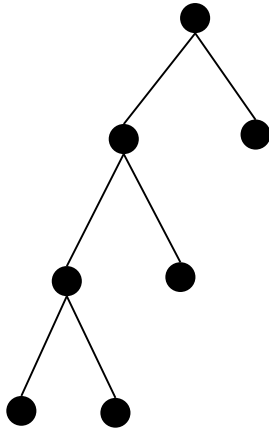
Indique cuáles de los siguientes árboles son  $n$ -arios completos y cuáles  $n$ -arios



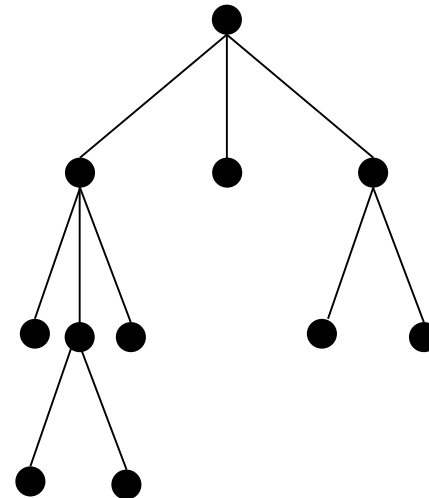
# Árboles

---

Indique cuáles de los siguientes árboles son  $n$ -arios completos y cuáles  $n$ -arios



Árbol binario  
completo

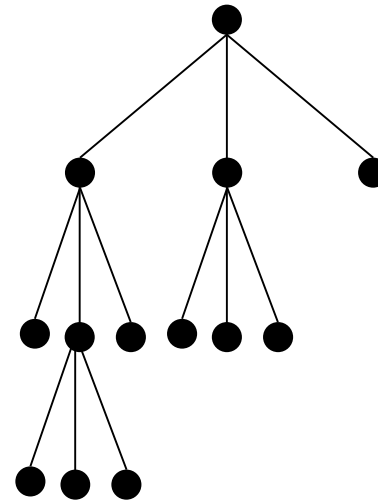
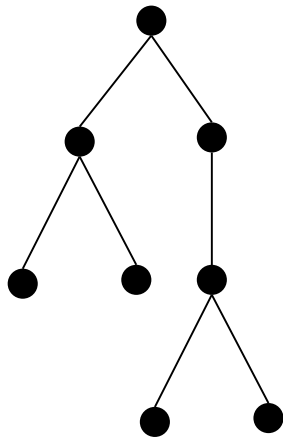


Árbol 3-ario

# Árboles

---

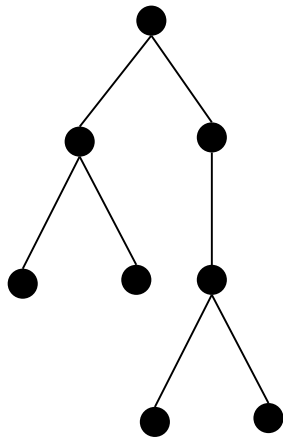
Indique cuáles de los siguientes árboles son  $n$ -arios completos y cuáles  $n$ -arios



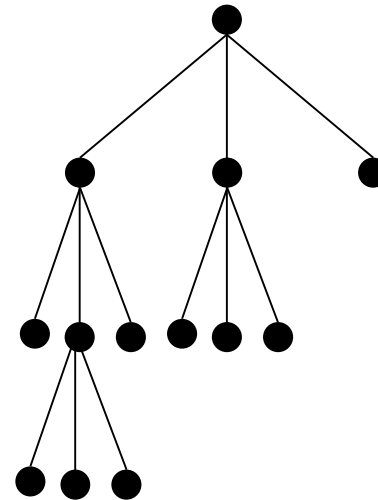
# Árboles

---

Indique cuáles de los siguientes árboles son  $n$ -arios completos y cuáles  $n$ -arios



Árbol binario



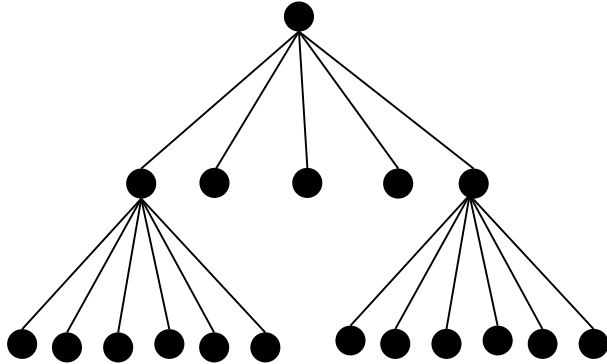
Árbol 3-ario  
completo



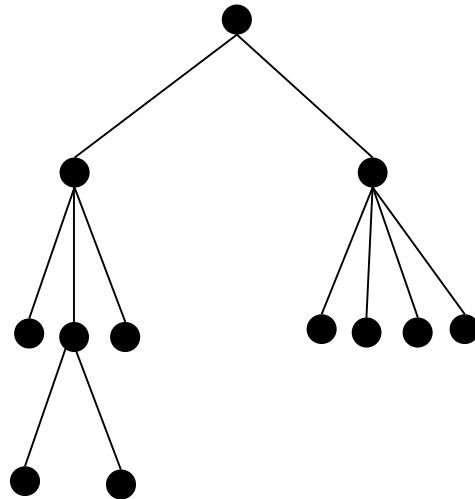
# Árboles

Indique cuáles de los siguientes árboles son  $n$ -arios completos y cuáles  $n$ -arios

6-ario

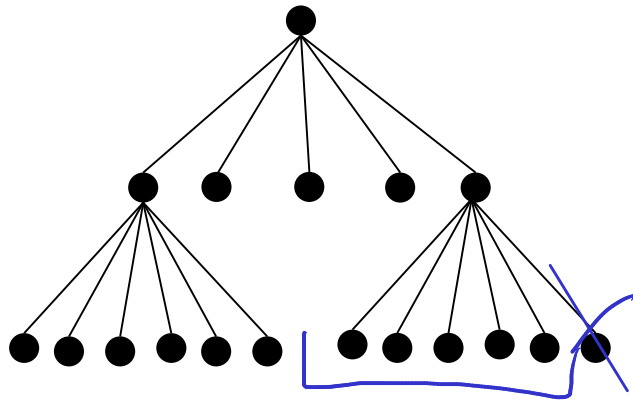


4-ario

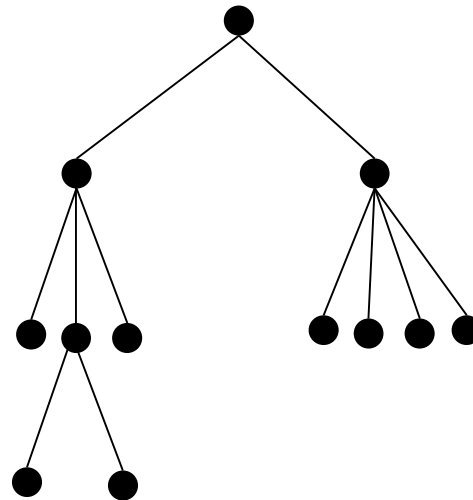


# Árboles

Indique cuáles de los siguientes árboles son  $n$ -arios completos y cuáles  $n$ -arios



Árbol 5-ario completo

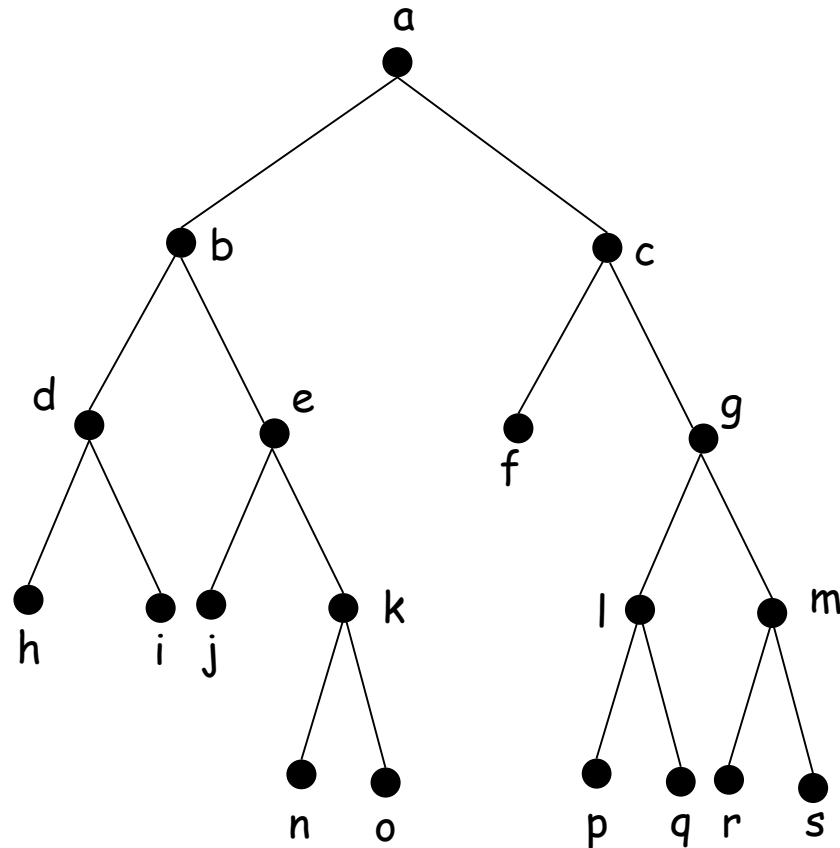


Árbol 4-ario

# Árboles

---

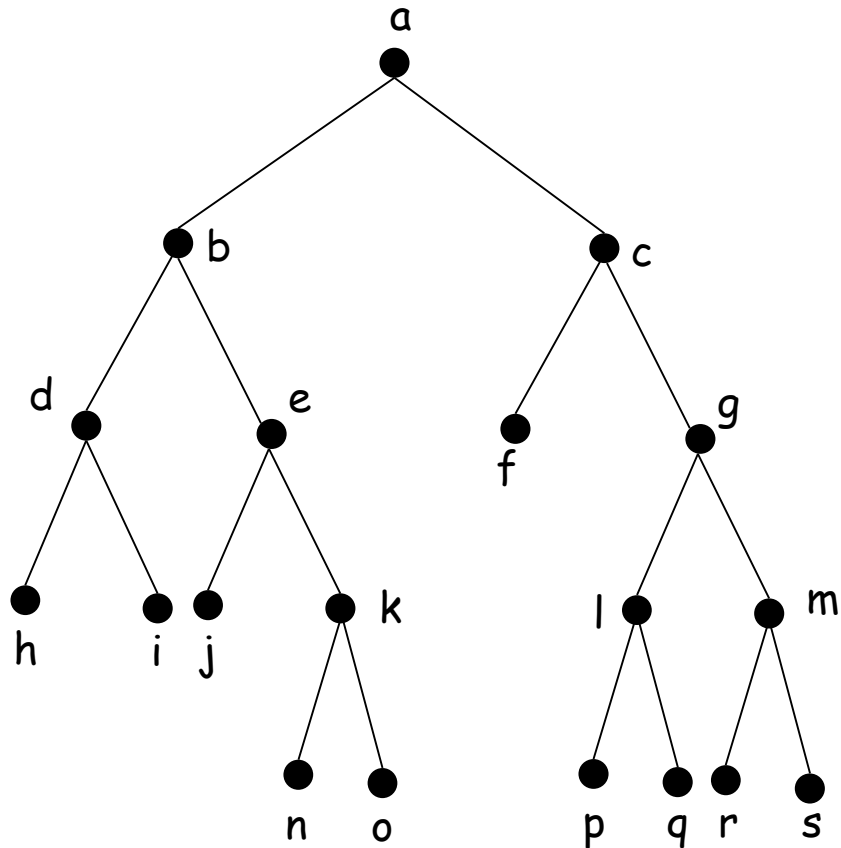
## Subárboles



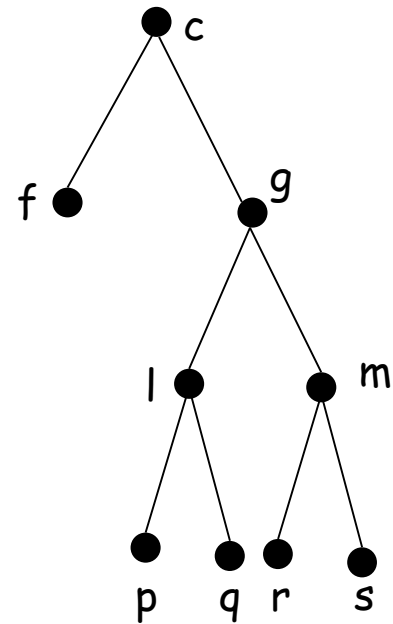
Árbol T

# Árboles

## Subárboles



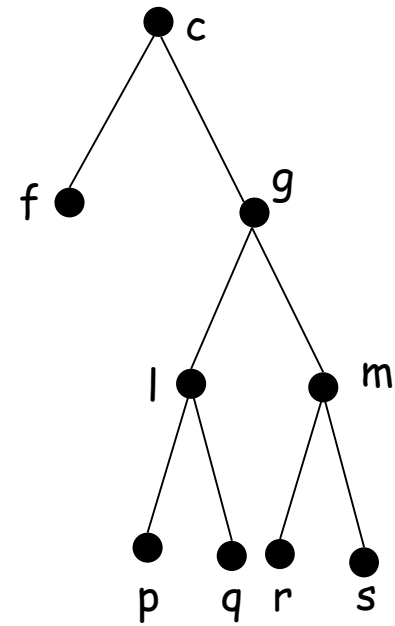
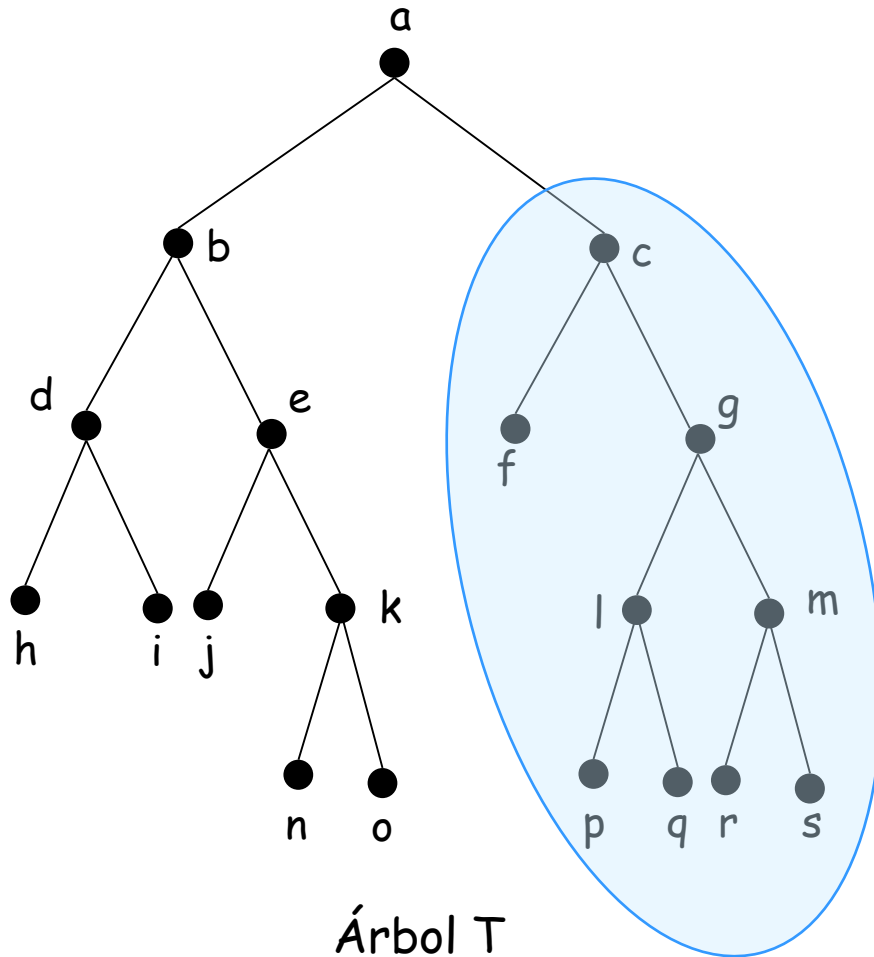
Árbol  $T$



Subárbol de  $T$  con raíz en  $c$

# Árboles

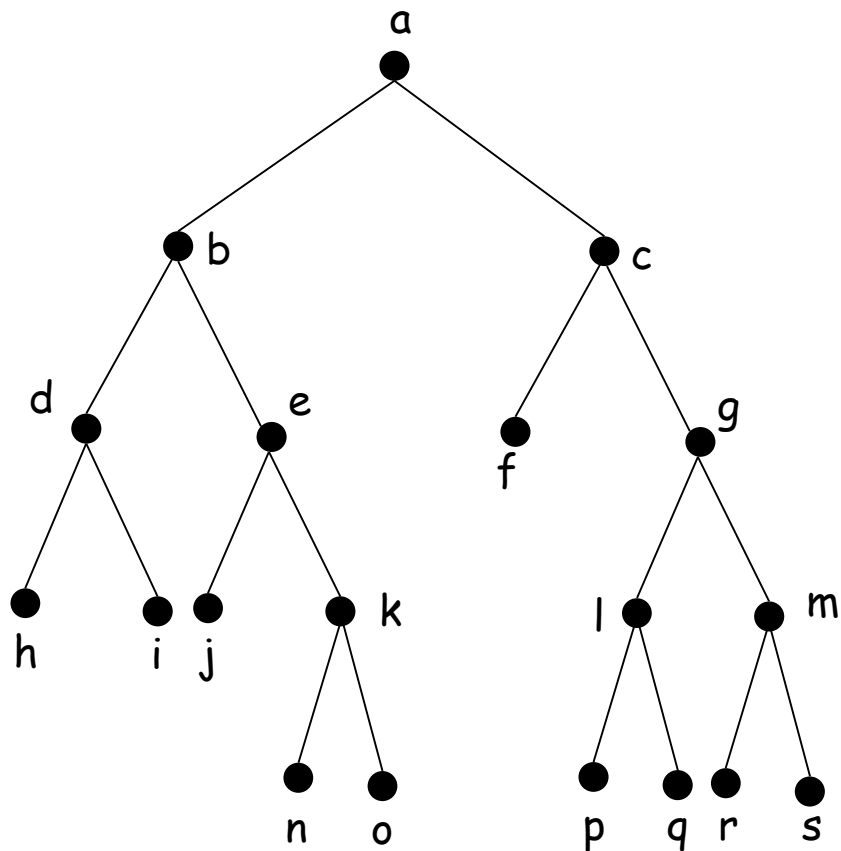
## Subárboles



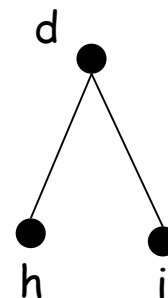
Subárbol de  $T$  con raíz en  $c$

# Árboles

## Subárboles



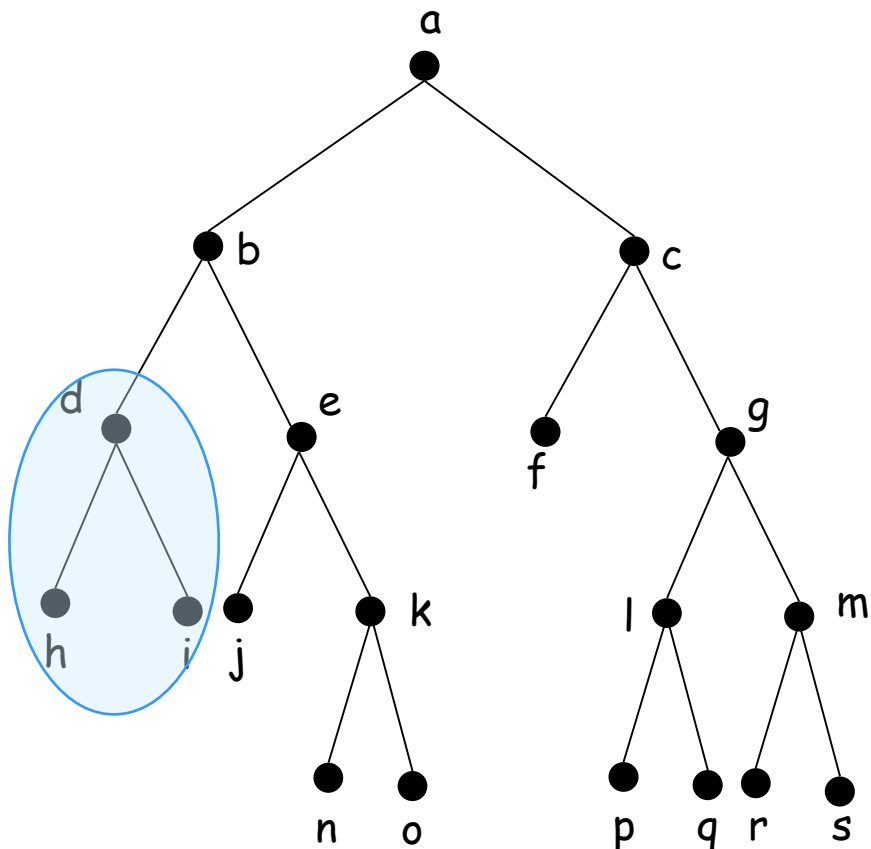
Árbol T



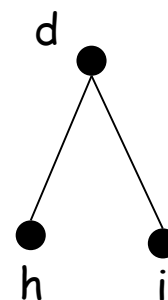
Subárbol de T con raíz en d

# Árboles

## Subárboles



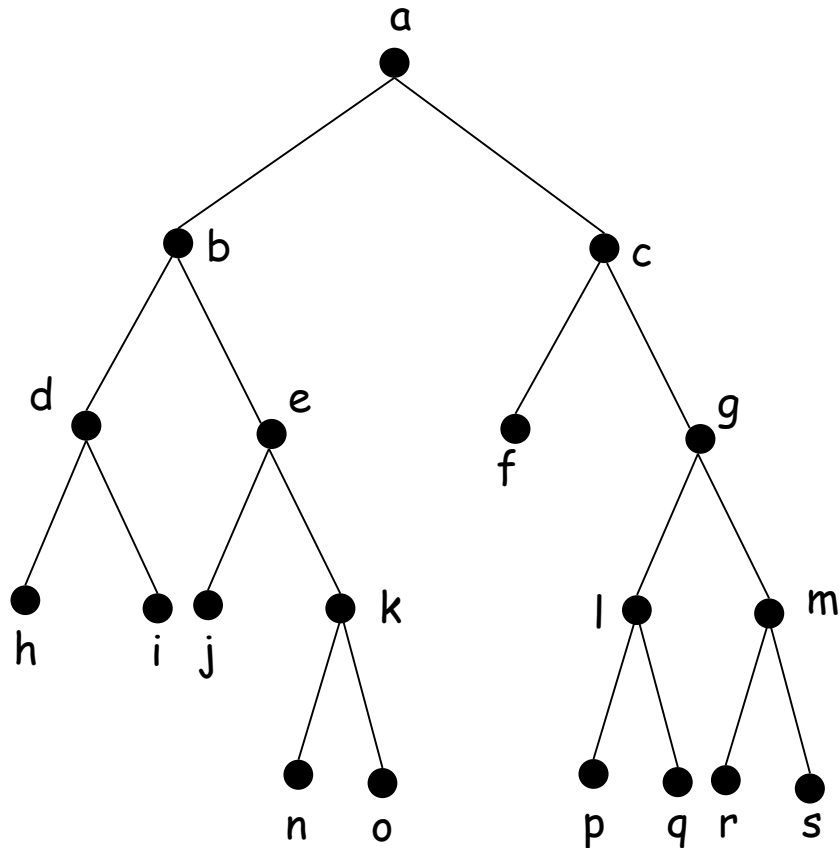
Árbol T



Subárbol de T con raíz en d

# Árboles

## Subárboles



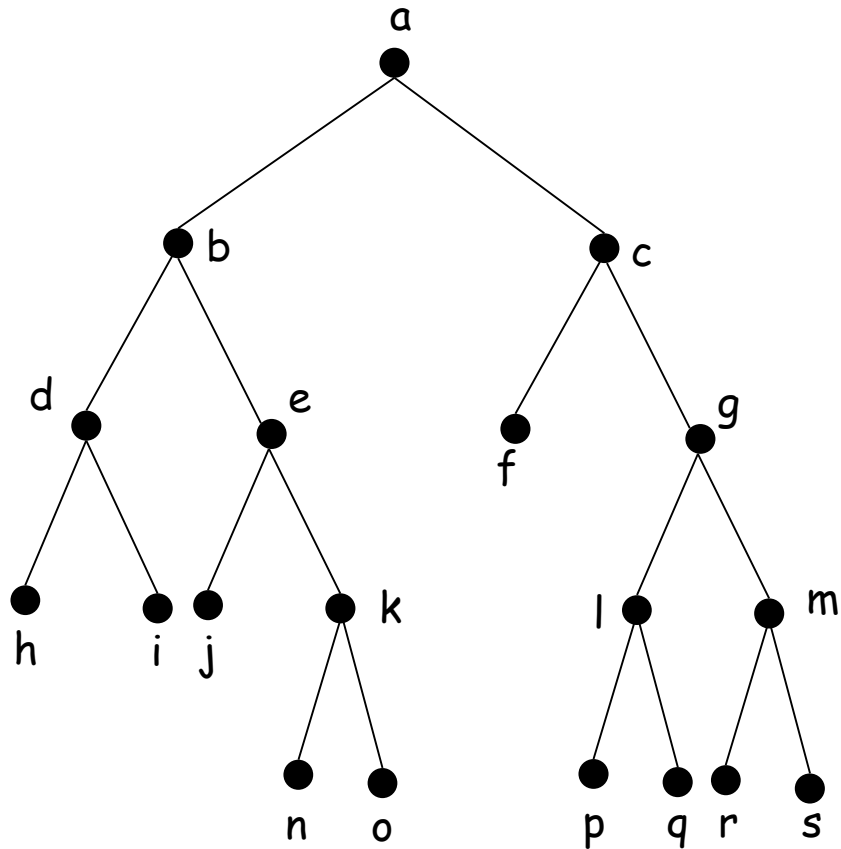
Árbol  $T$

Subárbol de  $T$  con raíz en  $b$

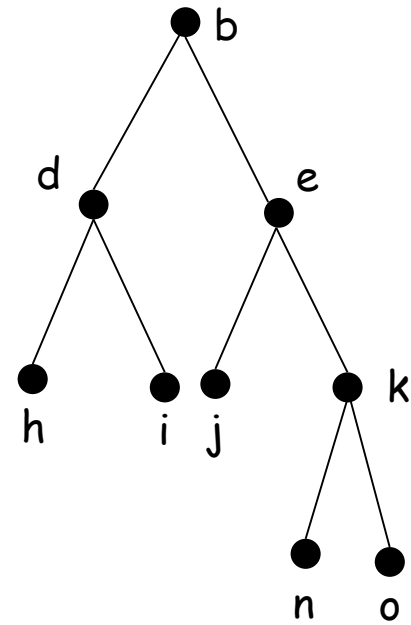


# Árboles

## Subárboles



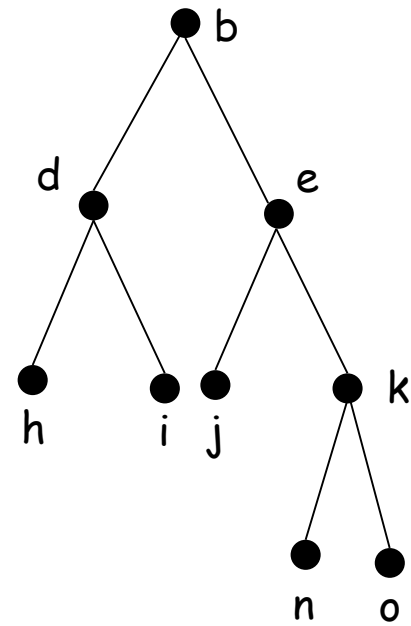
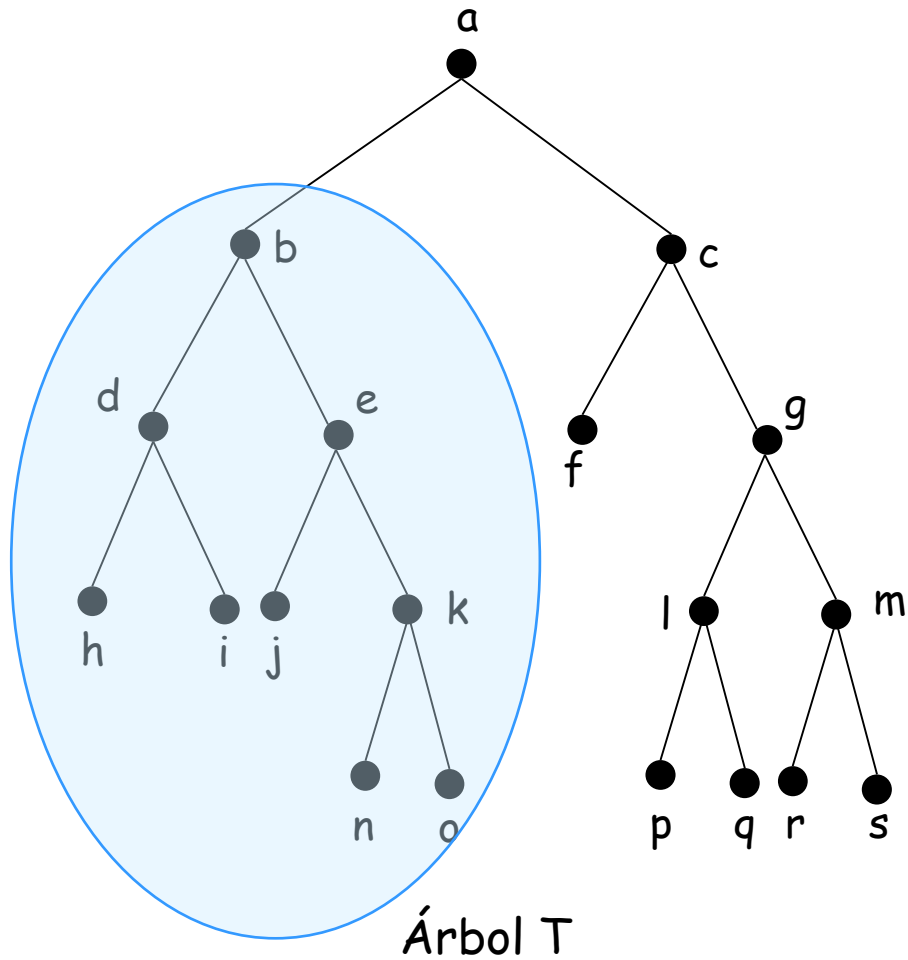
Árbol T



Subárbol de T con raíz en b

# Árboles

## Subárboles



Subárbol de  $T$  con raíz en  $b$

# Árboles

---

## Aplicaciones de los árboles

- Árboles de juego
- Árboles binarios de búsqueda
- Árboles de decisión

# Árboles

---

## Árboles de juego

- Se modelan los posibles movimientos de cada jugador en un juego con adversario
- Sirve para analizar el efecto de las jugadas

# Árboles

---

## Teoría de juegos

- Construir el árbol de juego
- Analizar el árbol

# Árboles

---

## Construir el árbol de juego

	x	o
x	x	o
	o	

- La jugada es de (X)

	X	O
X	X	O
	O	

	X	O
X	X	O
	O	

X	X	O
X	X	O
	O	

	X	O
X	X	O
X	O	

	X	O
X	X	O
	O	X



	X	O
X	X	O
	O	

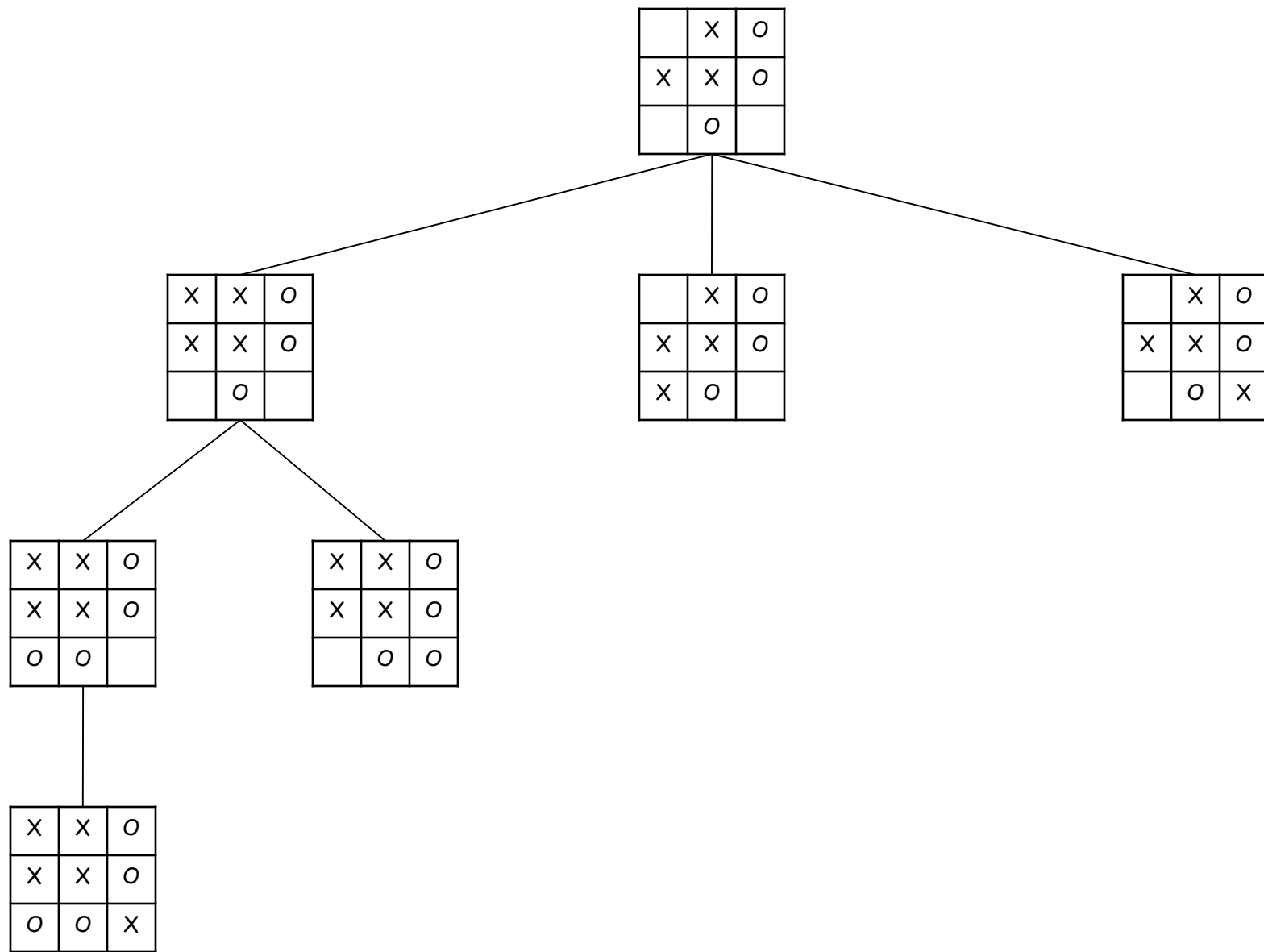
X	X	O
X	X	O
	O	

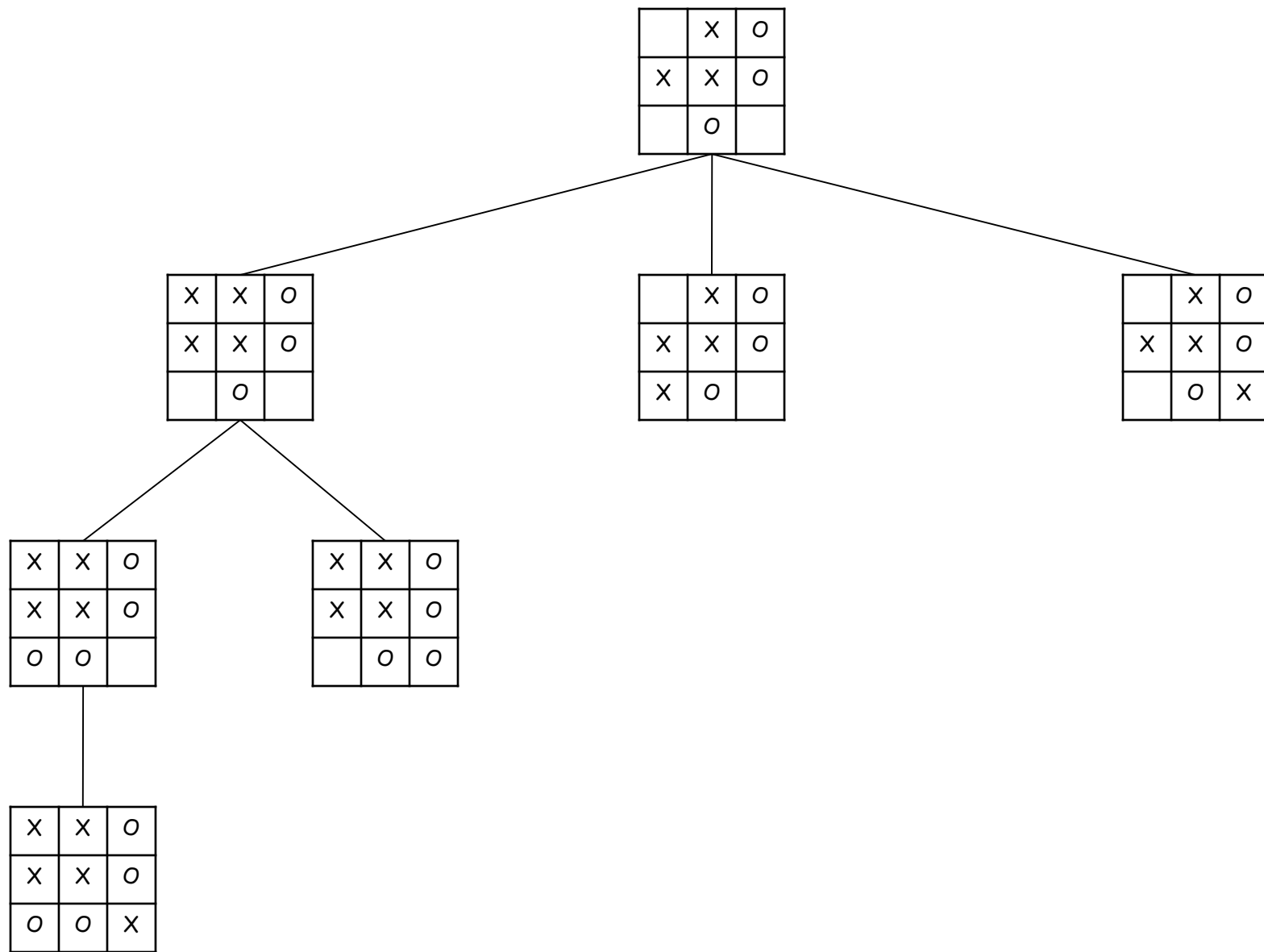
	X	O
X	X	O
X	O	

	X	O
X	X	O
	O	X

X	X	O
X	X	O
O	O	

X	X	O
X	X	O
	O	O





**Gana  
jugador1**

	X	O
X	X	O
	O	

X	X	O
X	X	O
	O	

	X	O
X	X	O
X	O	

	X	O
X	X	O
	O	X

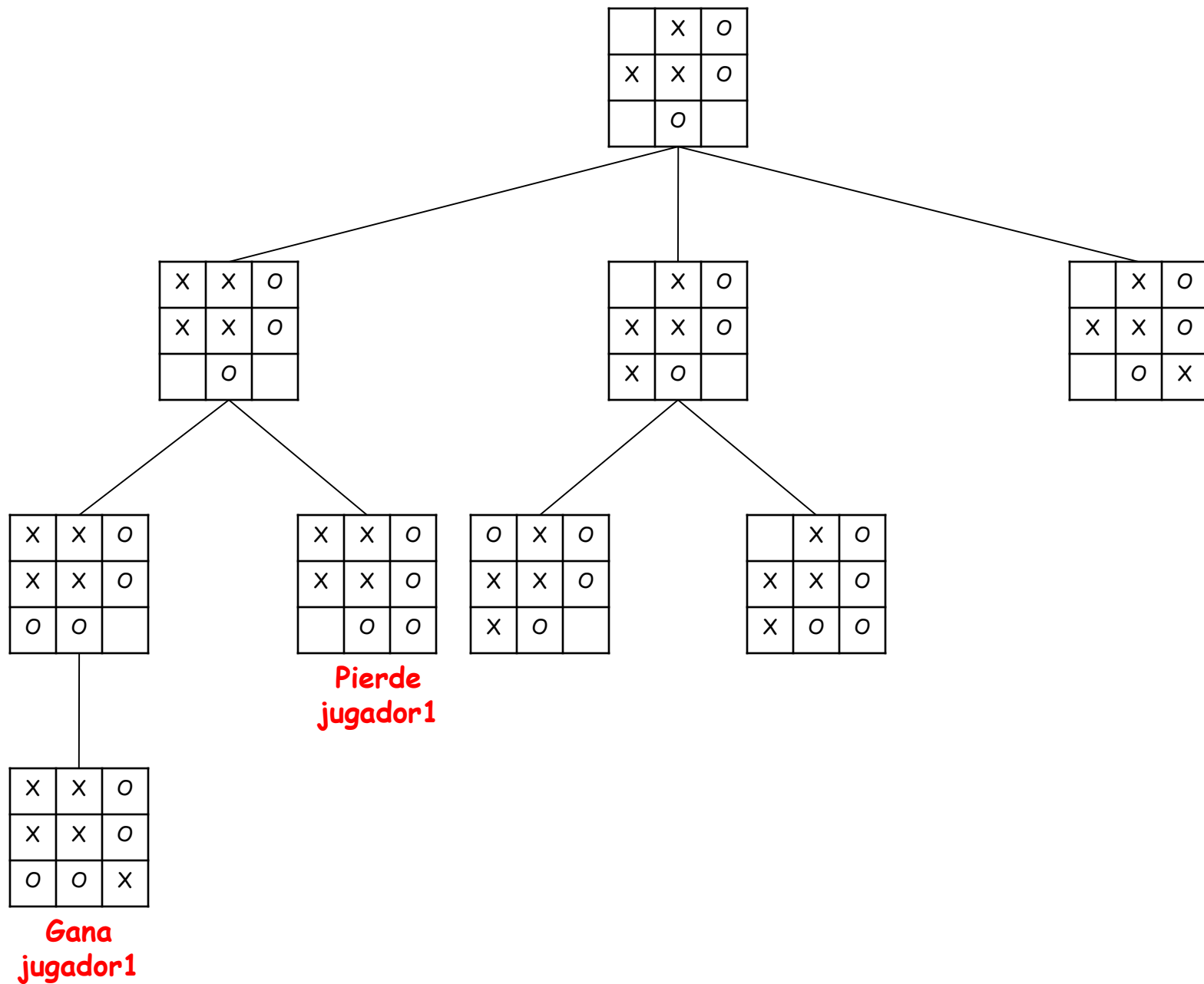
X	X	O
X	X	O
O	O	

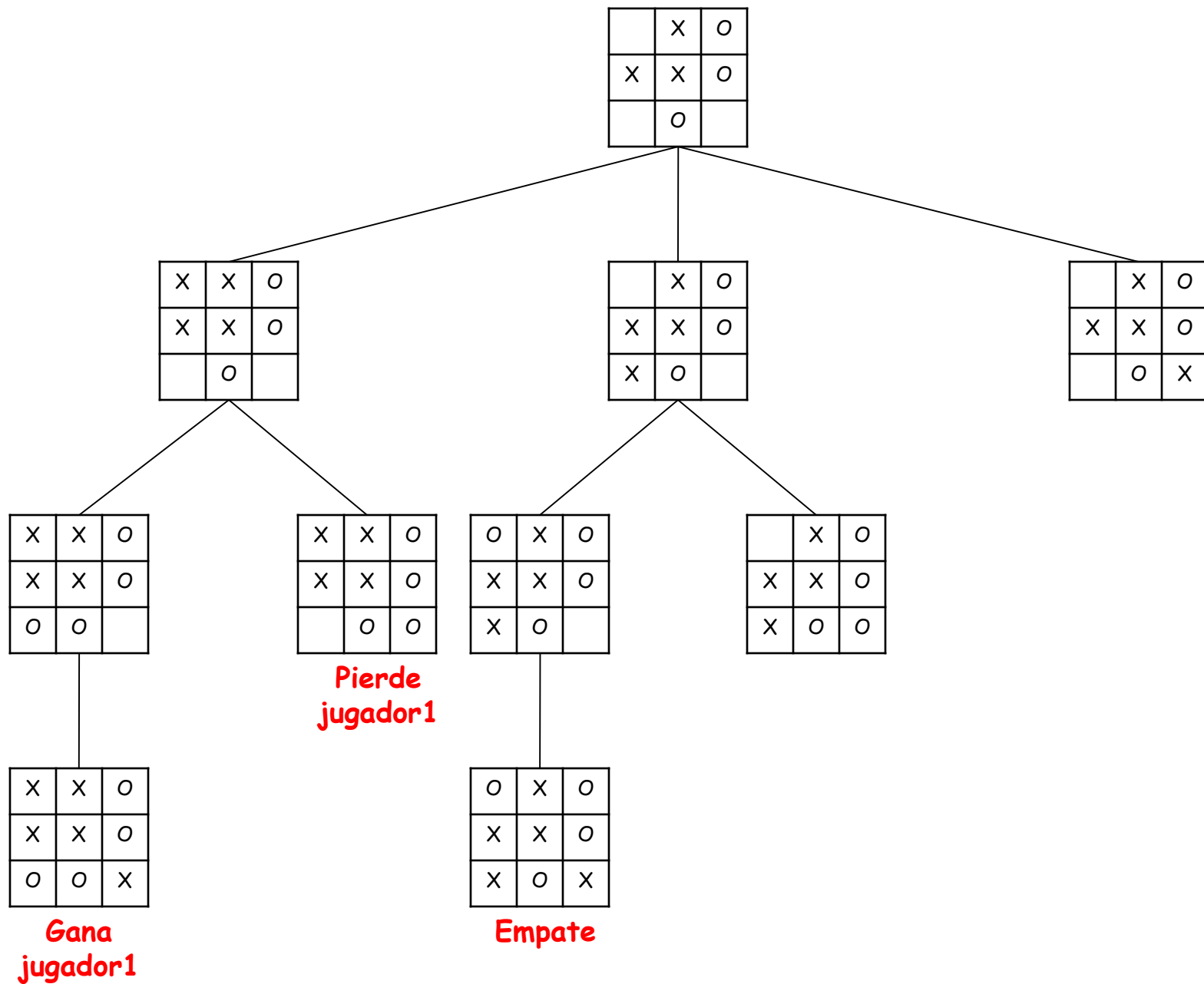
X	X	O
X	X	O
	O	O

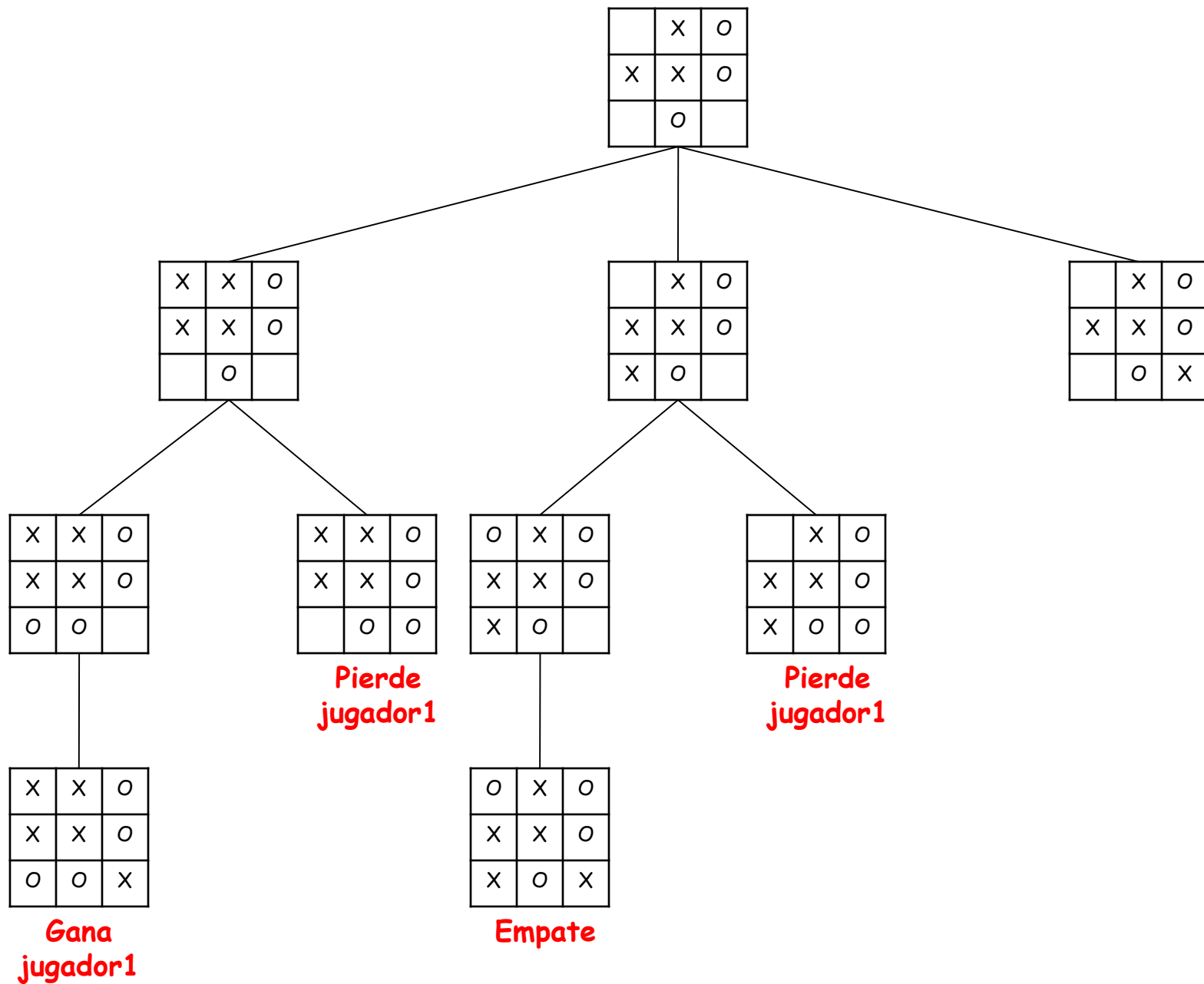
**Pierde  
jugador1**

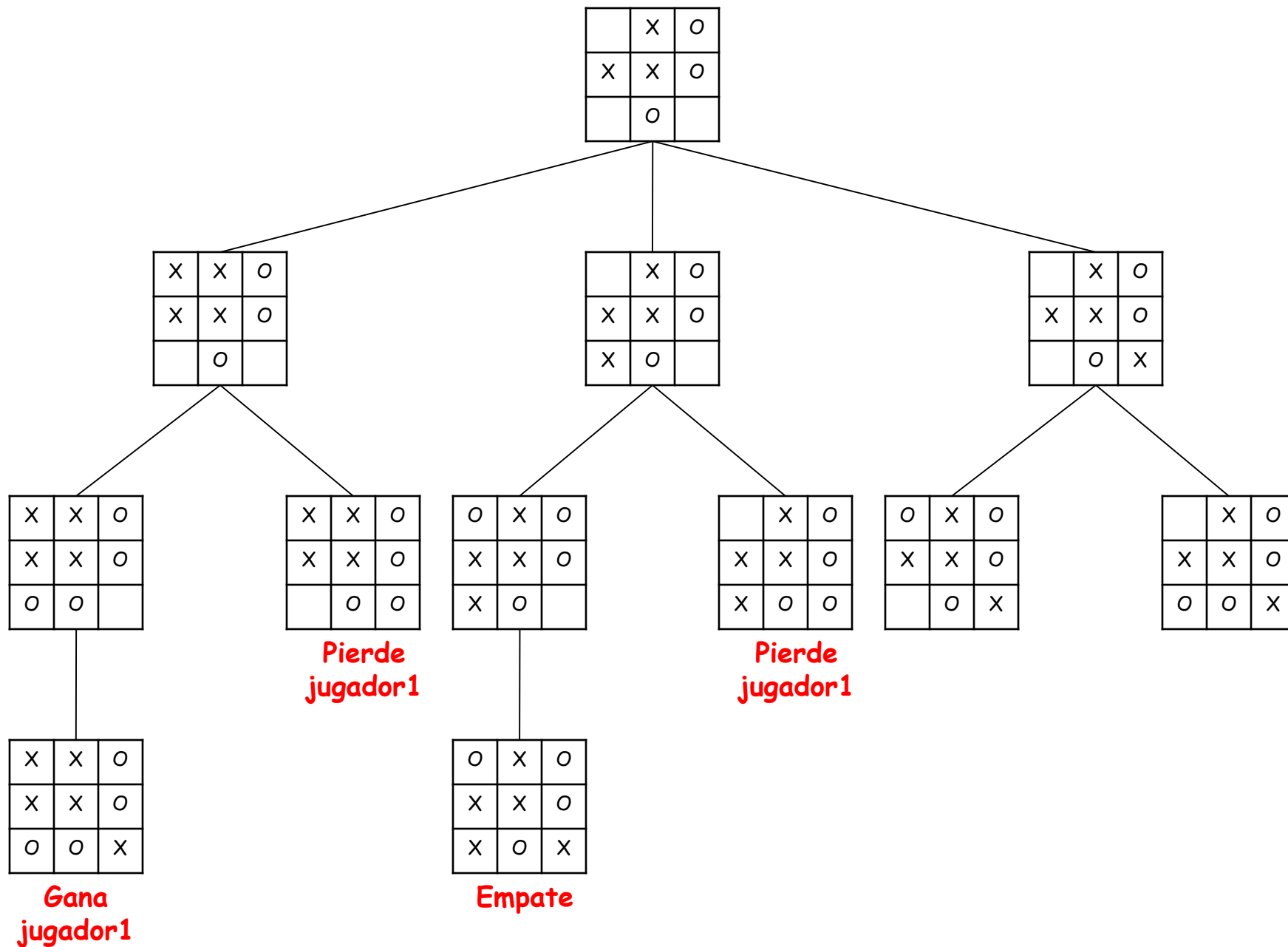
X	X	O
X	X	O
O	O	X

**Gana  
jugador1**

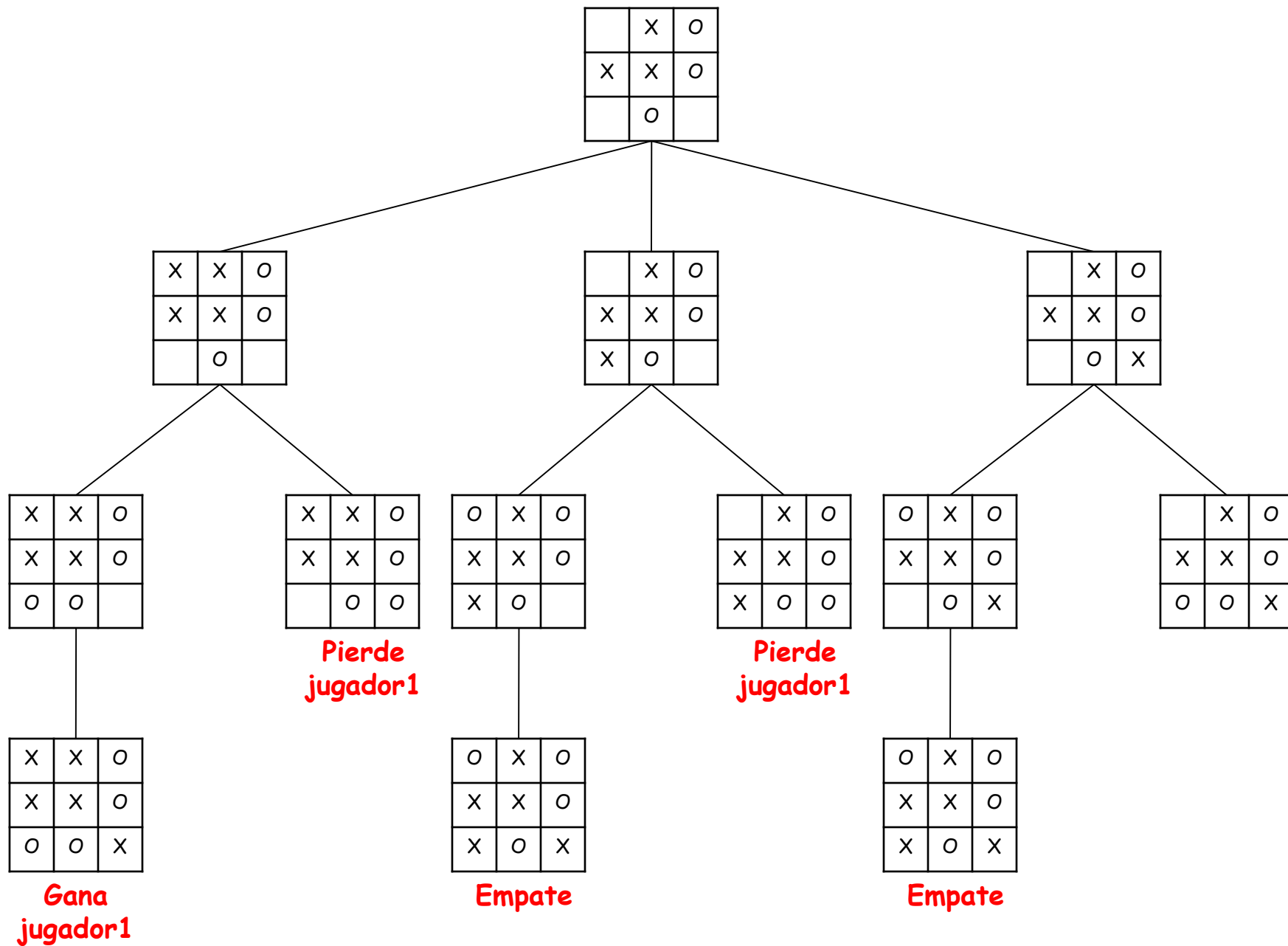


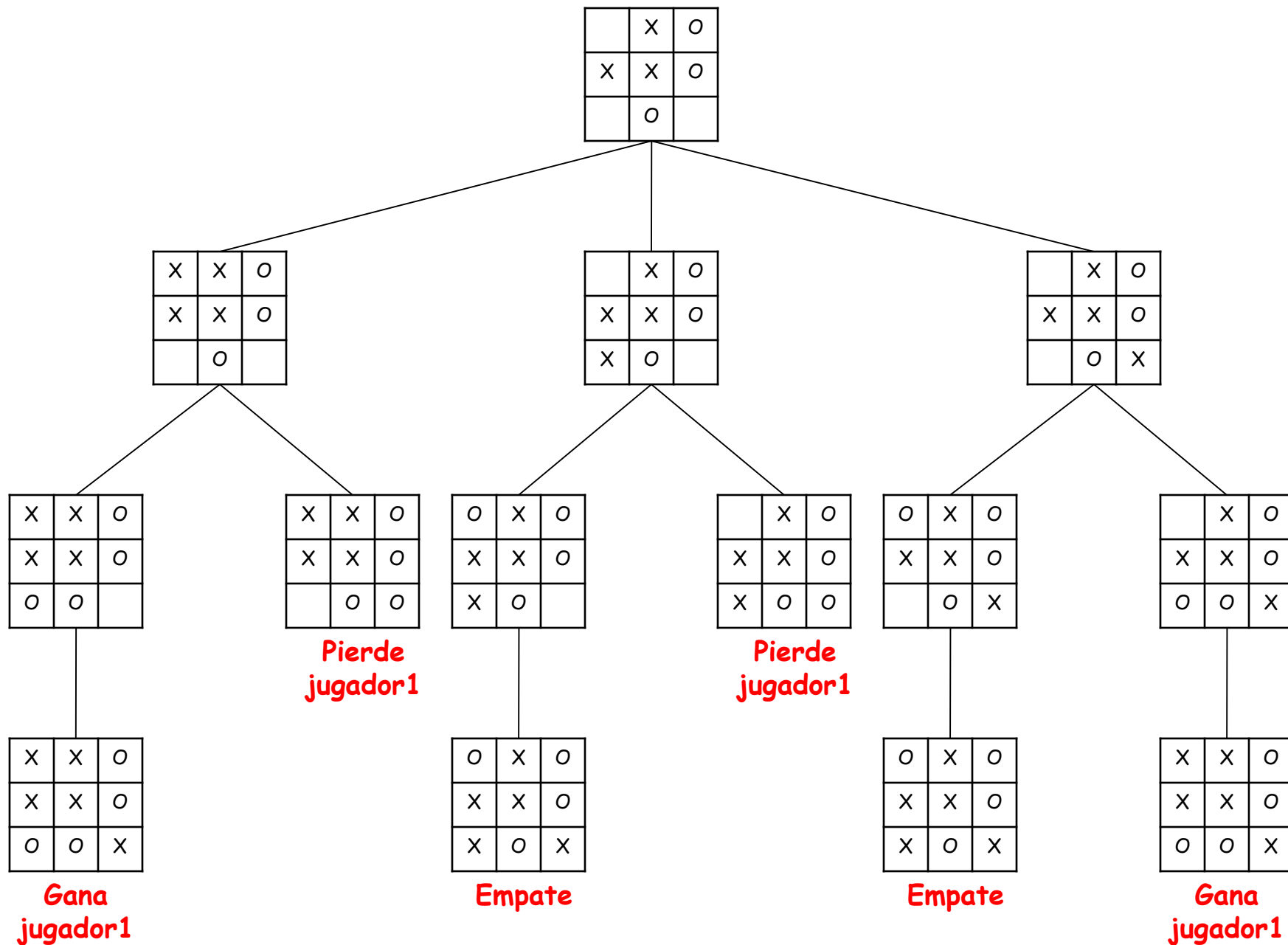




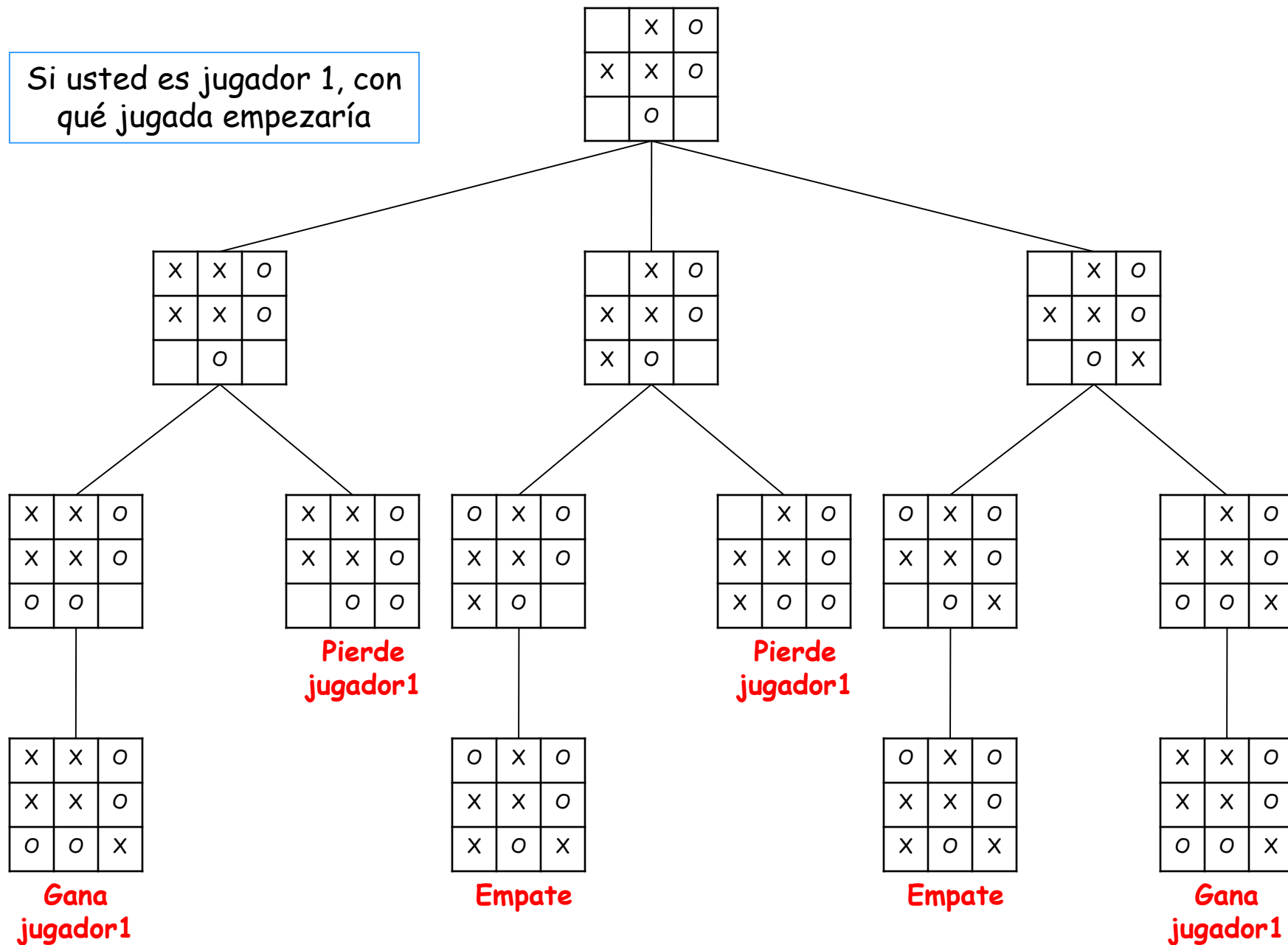








Si usted es jugador 1, con  
qué jugada empezaría



J1 ➡

	X	O
X	X	O
	O	

J2 ➡

X	X	O
X	X	O
	O	

	X	O
X	X	O
X	O	

	X	O
X	X	O
	O	X

J1 ➡

X	X	O
X	X	O
O	O	

X	X	O
X	X	O
	O	O

O	X	O
X	X	O
X	O	

	X	O
X	X	O
X	O	O

O	X	O
X	X	O
	O	X

	X	O
X	X	O
O	O	X

Pierde  
jugador1

Pierde  
jugador1

J2 ➡

X	X	O
X	X	O
O	O	X

O	X	O
X	X	O
X	O	X

O	X	O
X	X	O
X	O	X

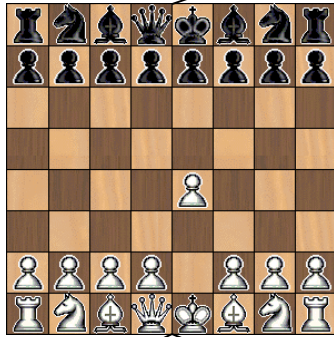
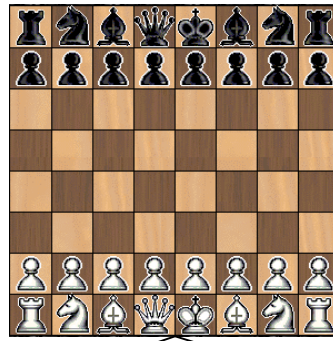
X	X	O
X	X	O
O	O	X

Gana  
jugador1

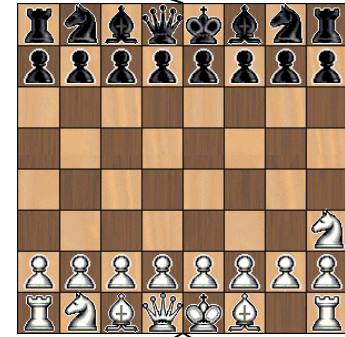
Empate

Empate

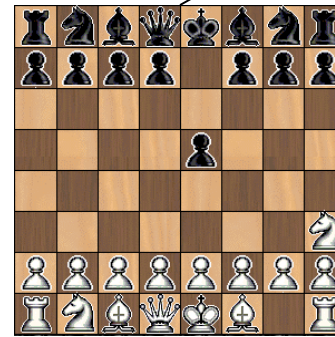
Gana  
jugador1



...



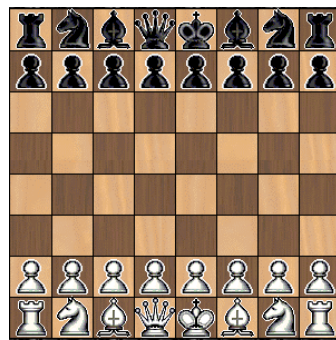
...



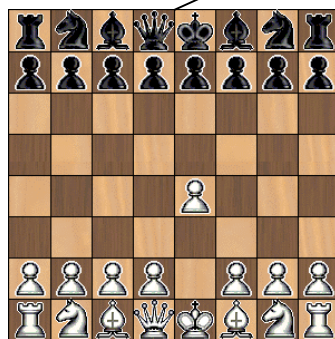
...



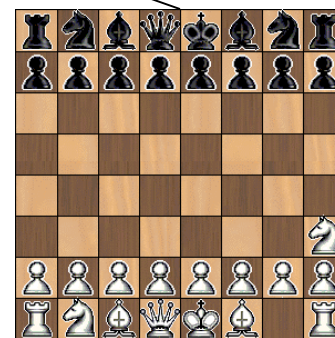
Jugador1 →



Jugador2 →



...



Jugador1 →



...



...



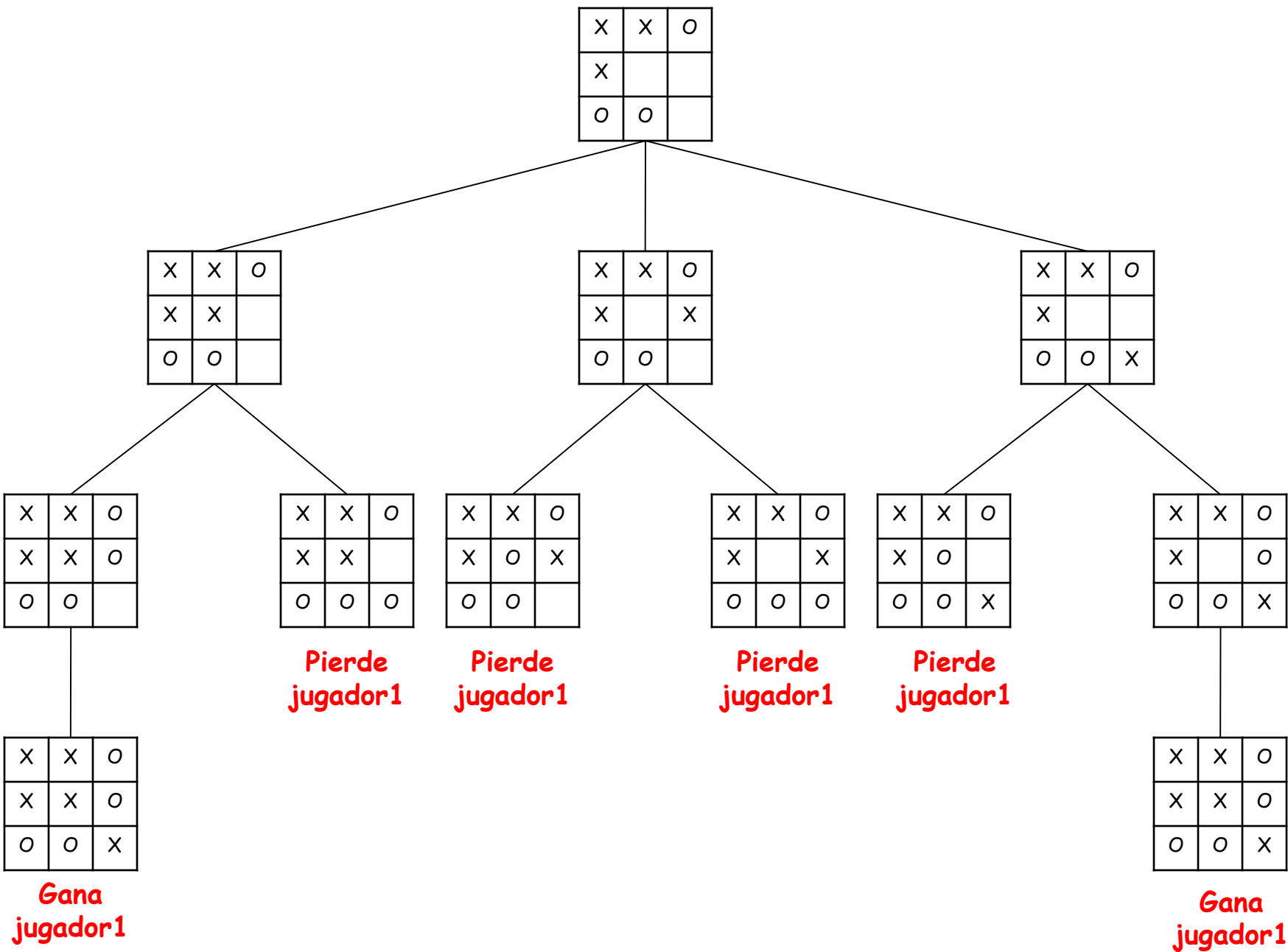
# Árboles

---

## Construir el árbol de juego

x	x	o
x		
o	o	

- La jugada es de (X)





Si usted es jugador 1, con  
qué jugada empezaría

X	X	O
X		
O	O	

X	X	O
X	X	
O	O	

X	X	O
X		X
O	O	

X	X	O
X		
O	O	X

X	X	O
X	X	O
O	O	

X	X	O
X	X	
O	O	O

X	X	O
X	O	X
O	O	

X	X	O
X		X
O	O	O

X	X	O
X	O	
O	O	X

X	X	O
X		O
O	O	X

X	X	O
X	X	O
O	O	X

**Gana  
jugador1**

**Pierde  
jugador1**

**Pierde  
jugador1**

**Pierde  
jugador1**

**Pierde  
jugador1**

X	X	O
X	X	O
O	O	X

**Gana  
jugador1**

# Árboles

---

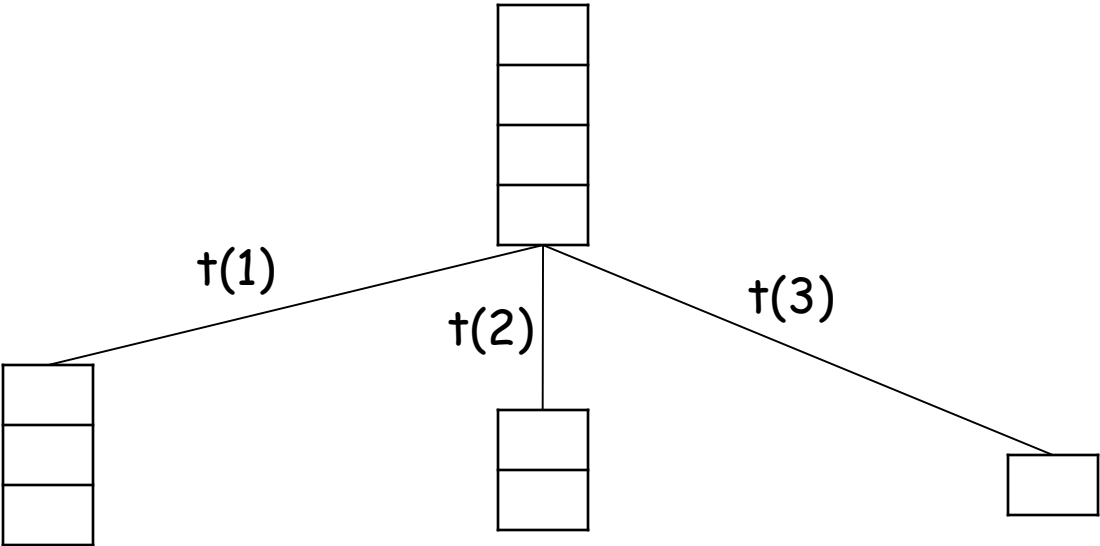
## Construir el árbol de juego



- **El juego del NIM.** Se tiene una pila de 4 fichas de la cual cada jugador puede tomar 1, 2 ó 3. El objetivo de cada jugador es obligar a su adversario a tomar la última ficha. Como los elementos están apilados, solo se pueden tomar fichas de su tope

J1 →

J2 →



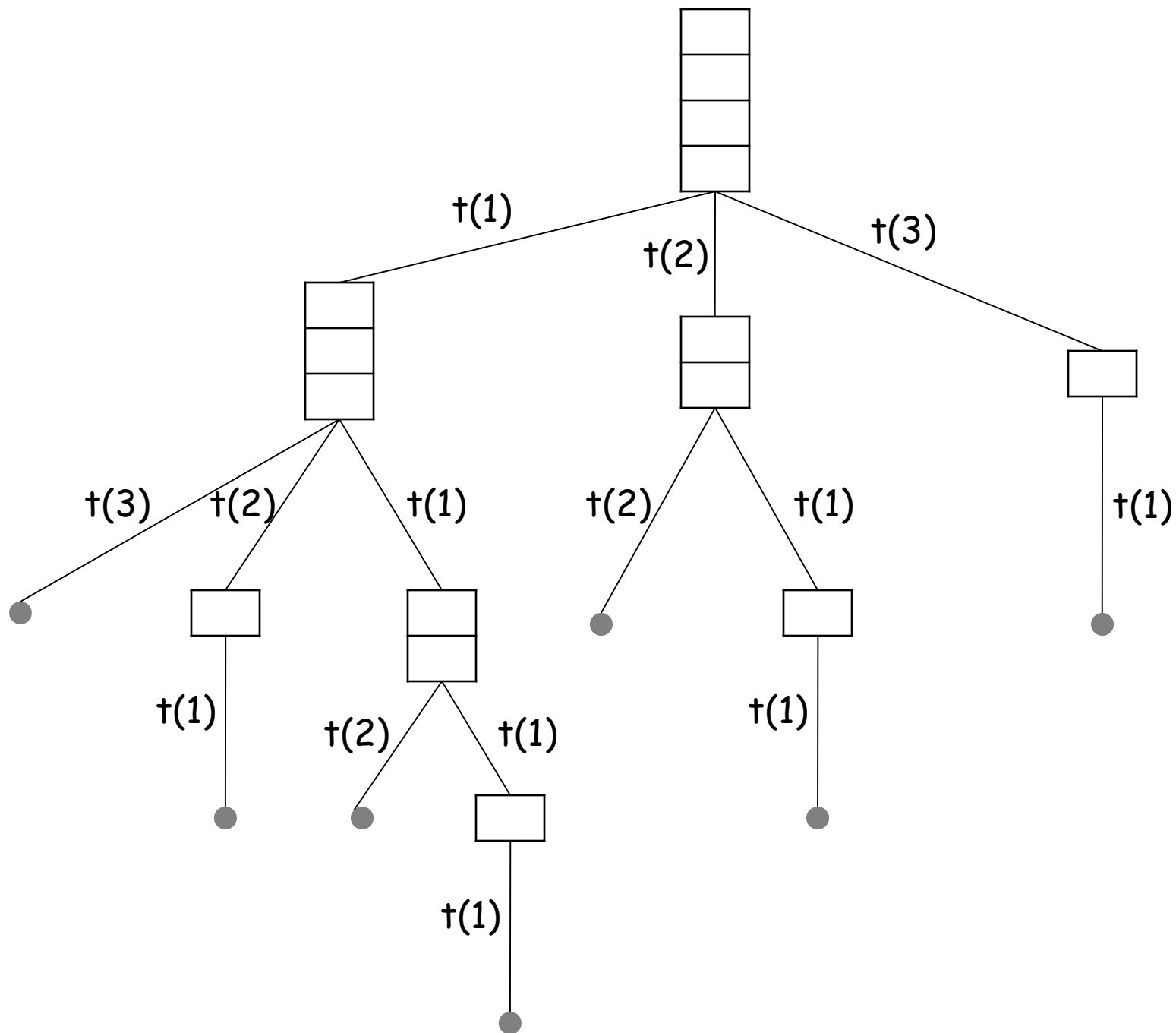
J1 →

J2 →

J1 →

J2 →

J1 →



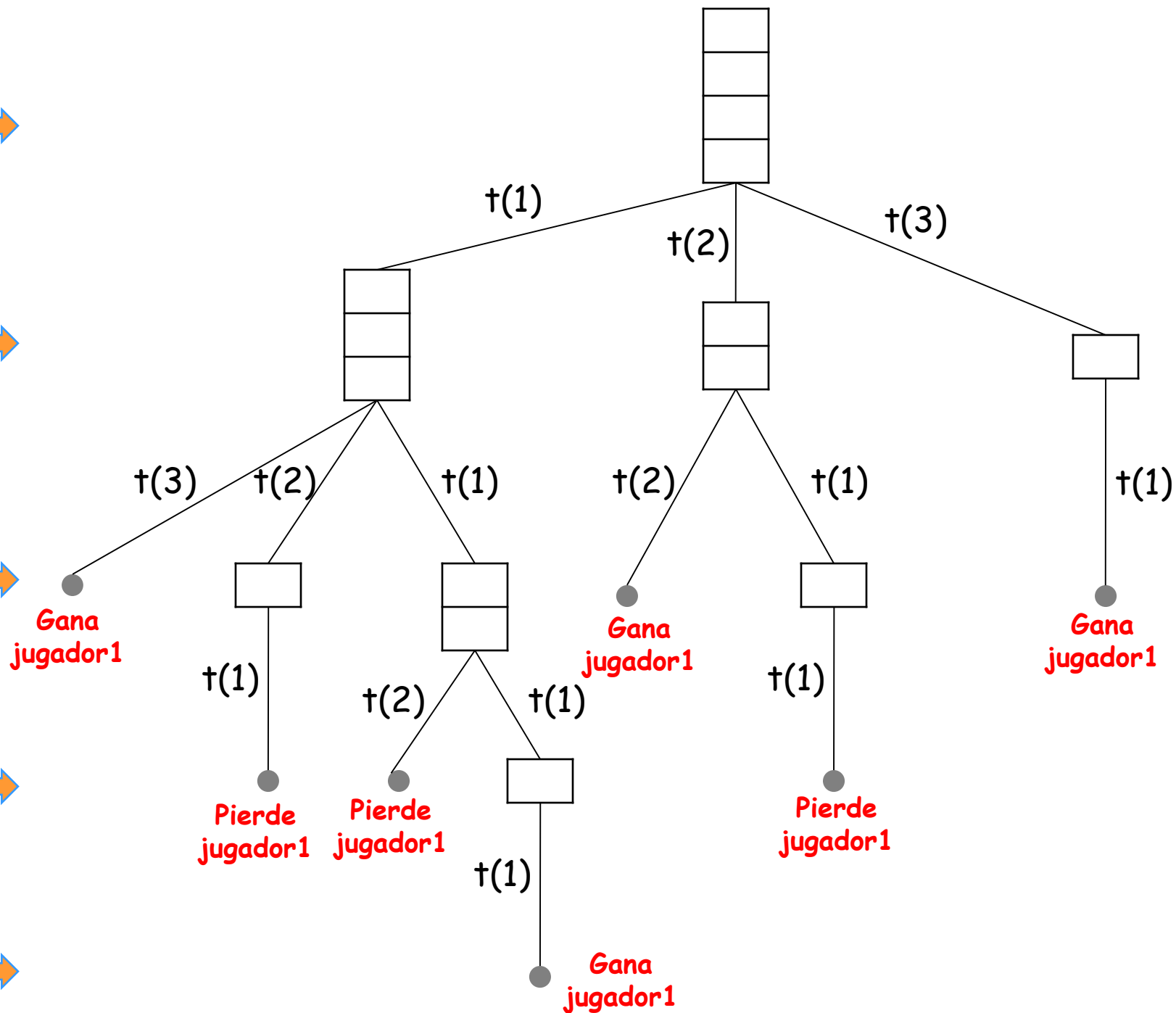
J1 →

J2 →

J1 →

J2 →

J1 →



# Árboles

---

## Aplicaciones de los árboles

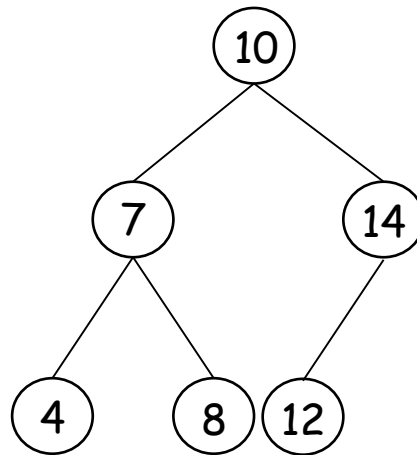
- Árboles de juego
- Árboles binarios de búsqueda
- Árboles de decisión

# Árboles

---

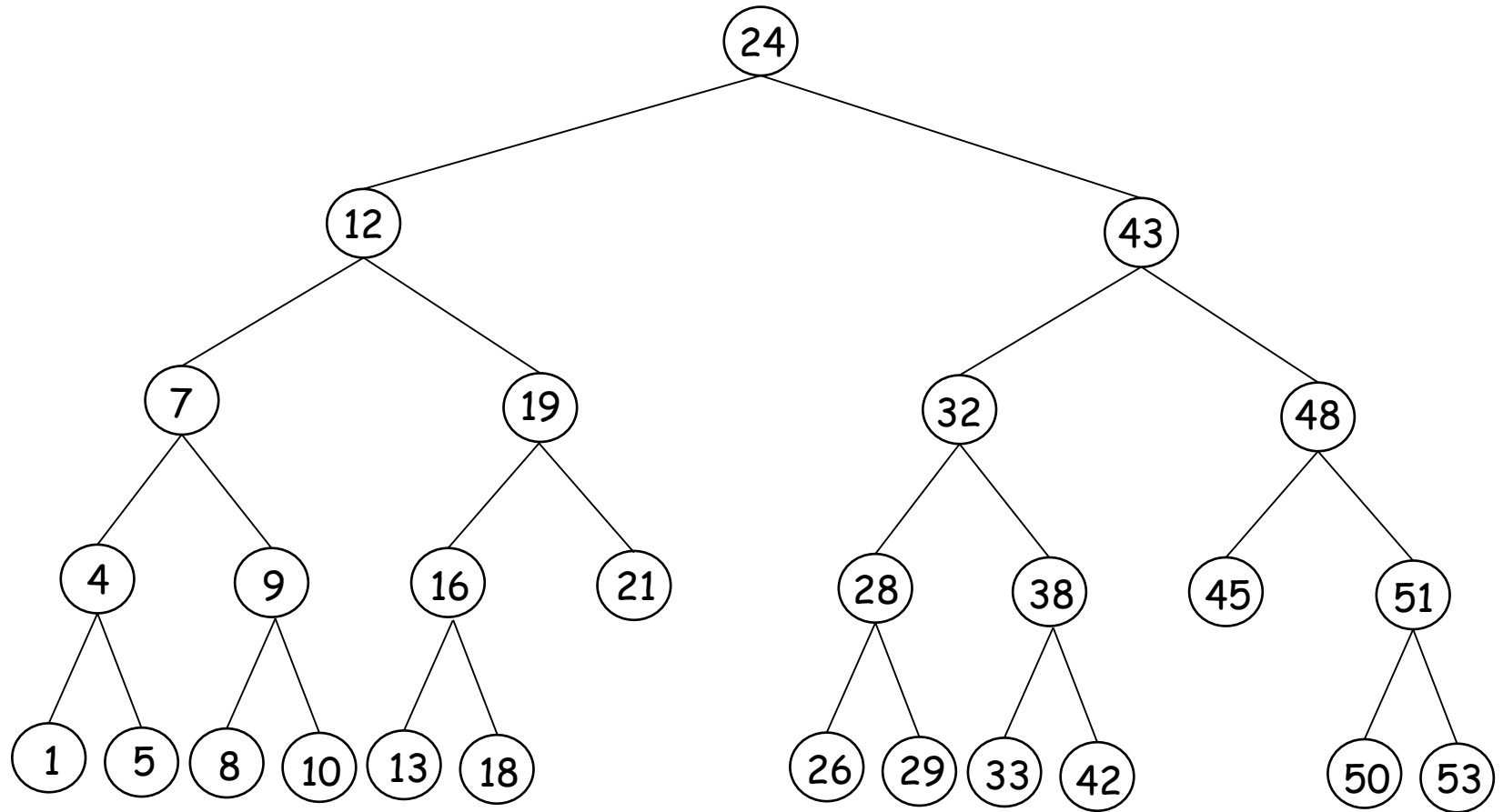
## Árboles binarios de búsqueda

Es un árbol binario en el que cada vértice tiene una llave. La llave de un vértice es mayor que las llaves de los vértices del subárbol izquierdo y menor que las llaves de los vértices del subárbol derecho



# Árboles

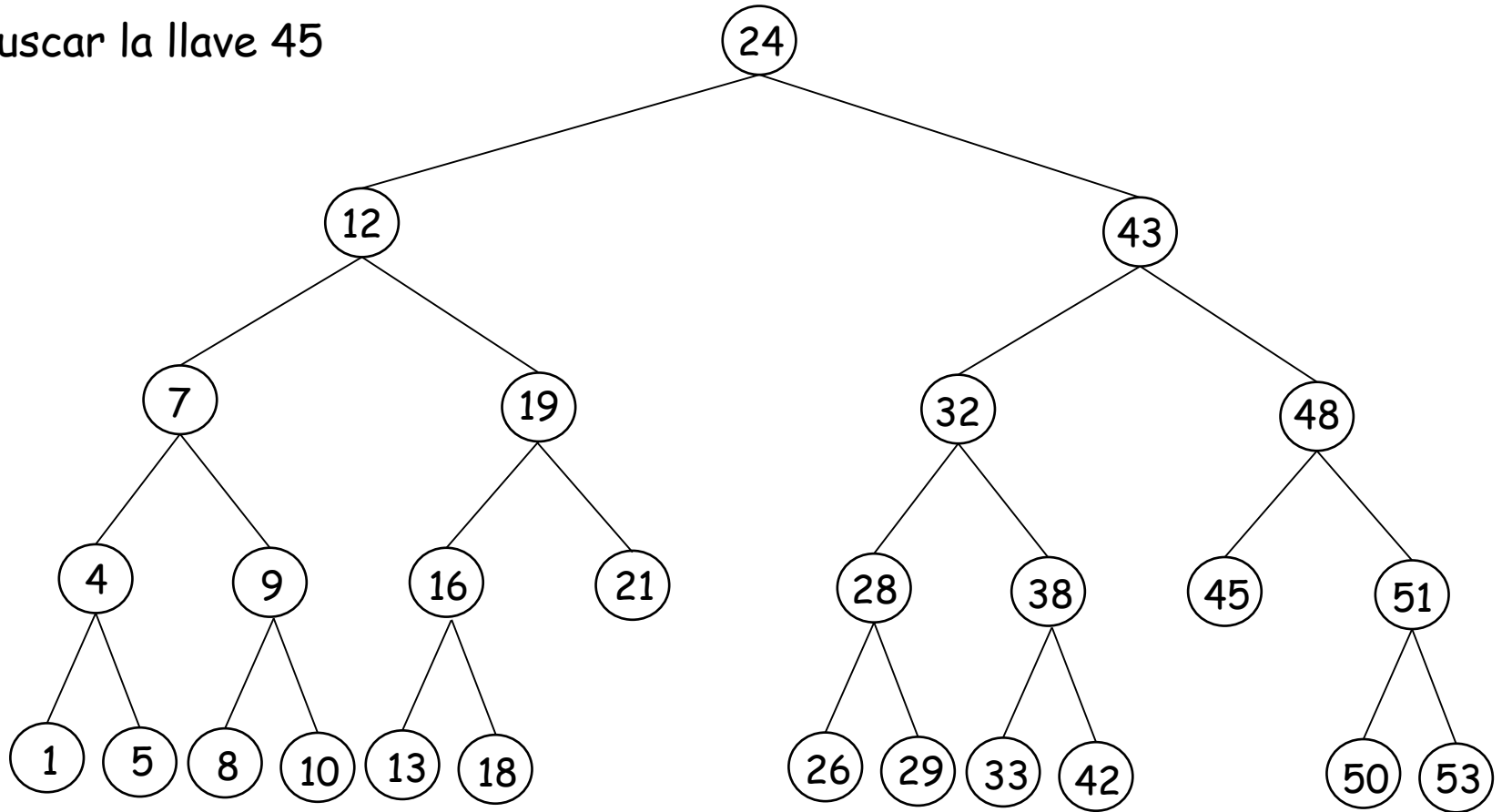
---



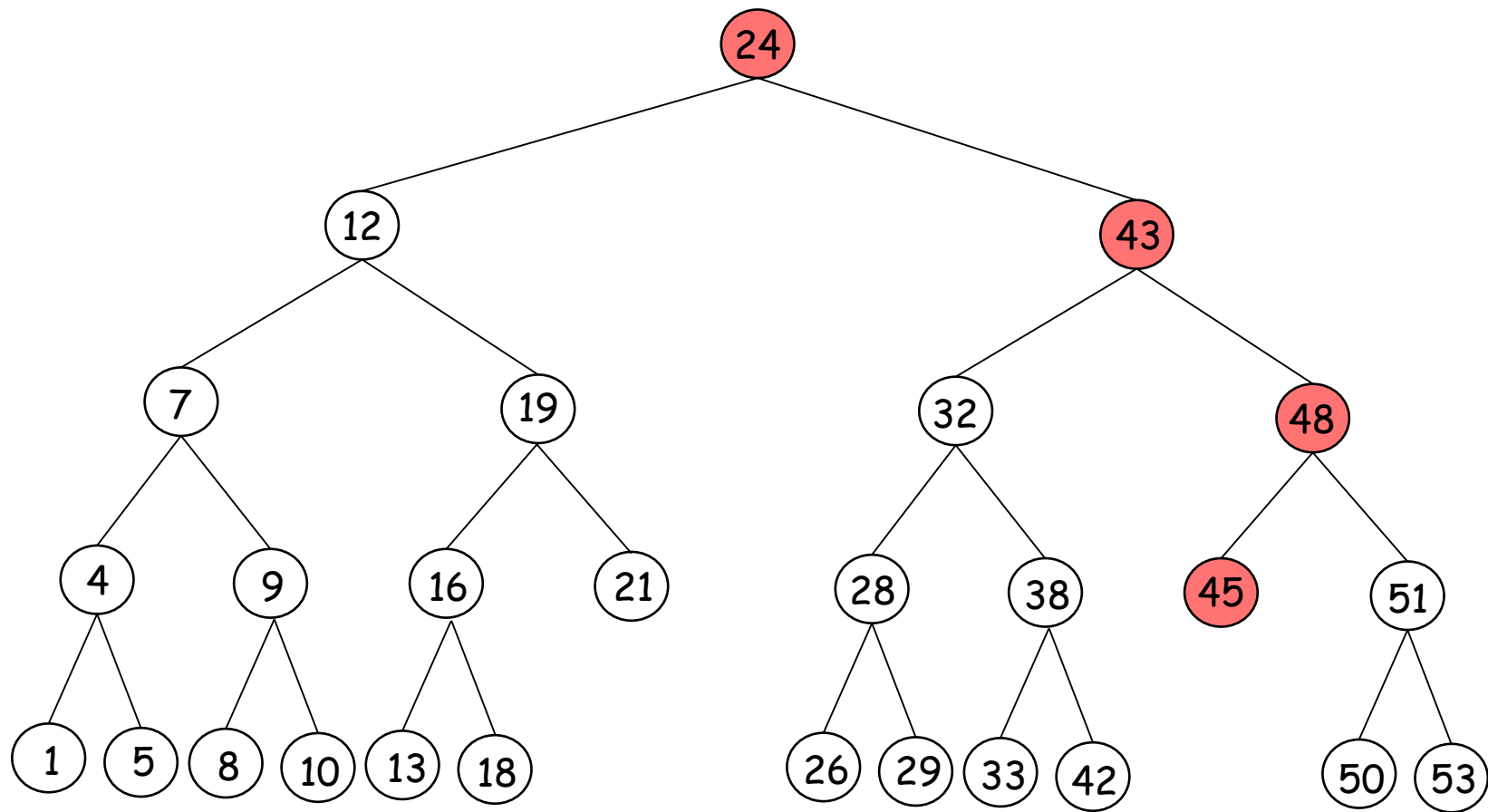


# Árboles

Buscar la llave 45



# Árboles

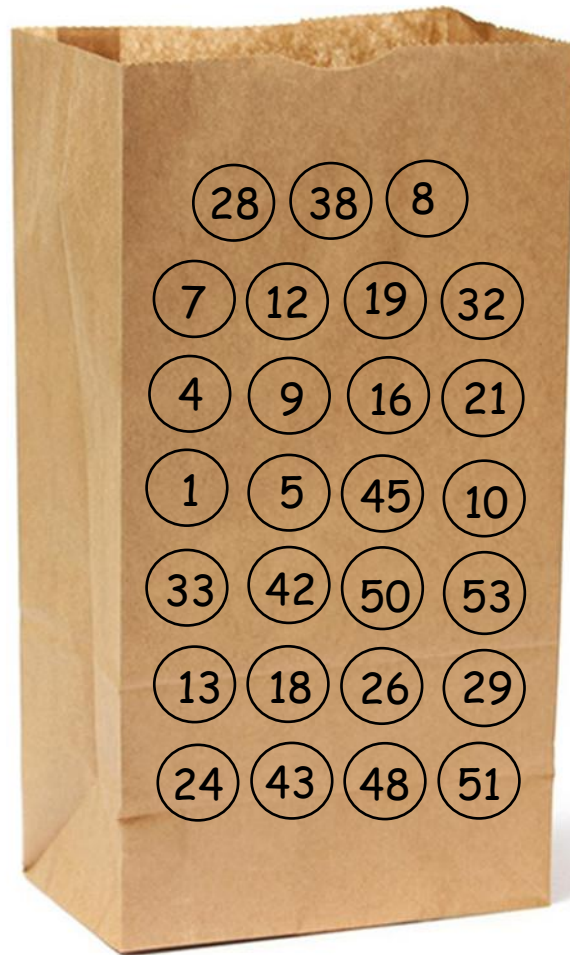


Se necesitan 4 comparaciones

# Árboles

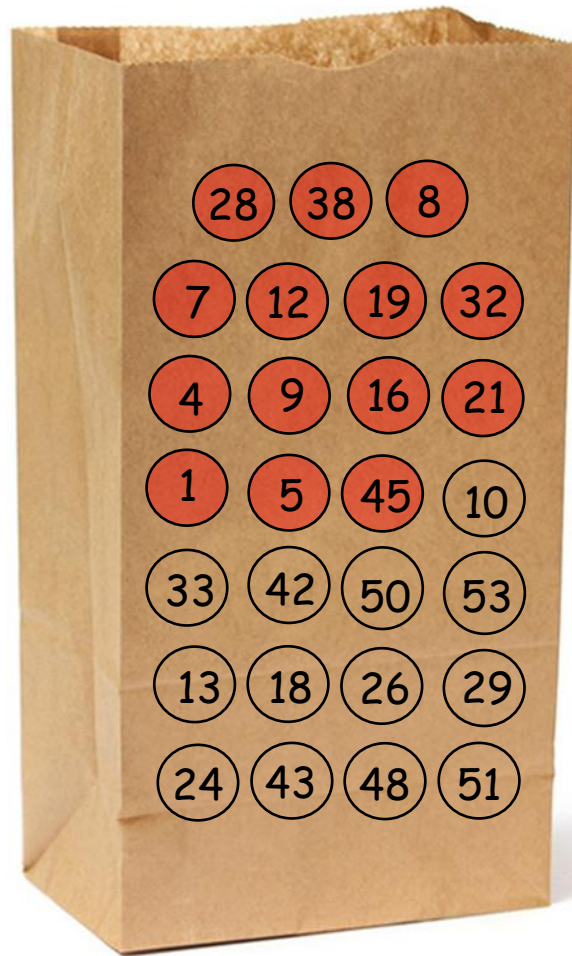
---

Buscar la llave 45



# Árboles

---

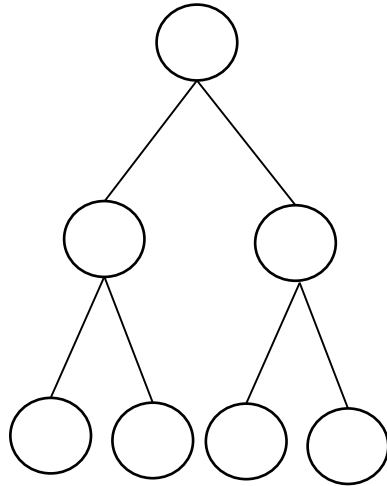


Se necesitan 14 comparaciones

# Árboles

---

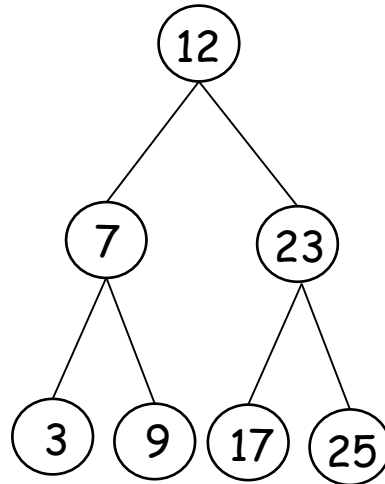
Crear un árbol de búsqueda binaria con las llaves 3,7,9,12,17,23,25 que tenga la siguiente estructura



# Árboles

---

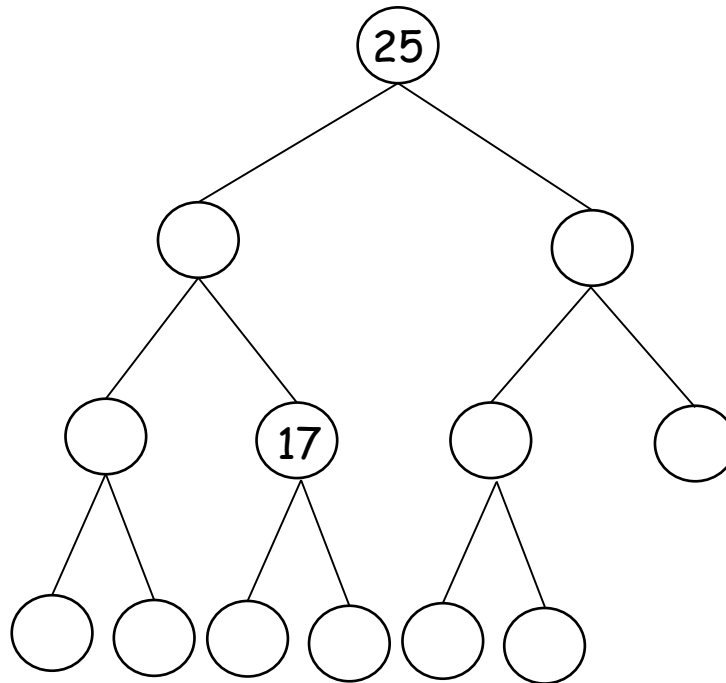
Crear un árbol de búsqueda binaria con las llaves 3,7,9,12,17,23,25 que tenga la siguiente estructura



# Árboles

---

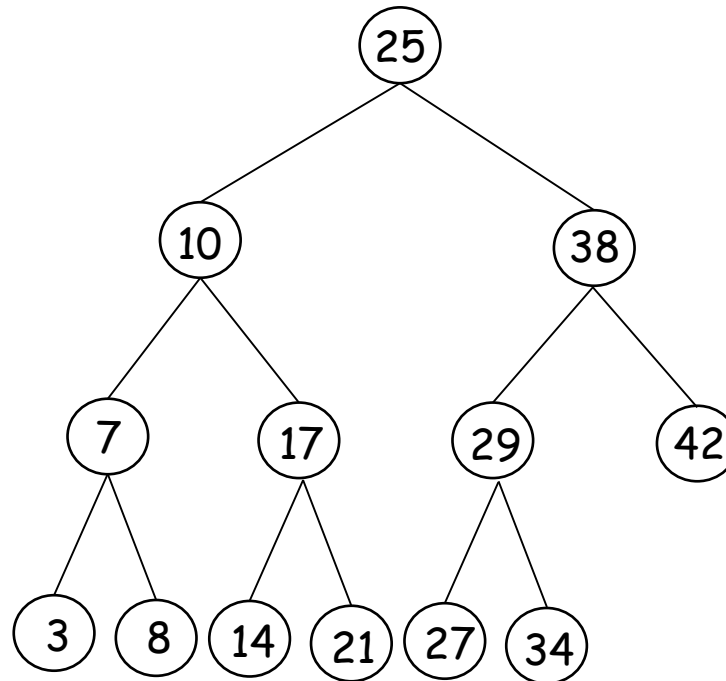
Crear un árbol de búsqueda binaria con las llaves  
3,7,8,10,14,17,21,25,27,29,34,38,42 que tenga la siguiente  
estructura



# Árboles

---

Crear un árbol de búsqueda binaria con las llaves  
3,7,8,10,14,17,21,25,27,29,34,38,42 que tenga la siguiente  
estructura





# Árboles

---

Crear un árbol de búsqueda binaria con las siguientes llaves:

- apple tv
- ipad
- iphone
- ipod touch
- nintendo wii
- playstation
- smart tv

# Árboles

---

apple tv

~~ipad~~

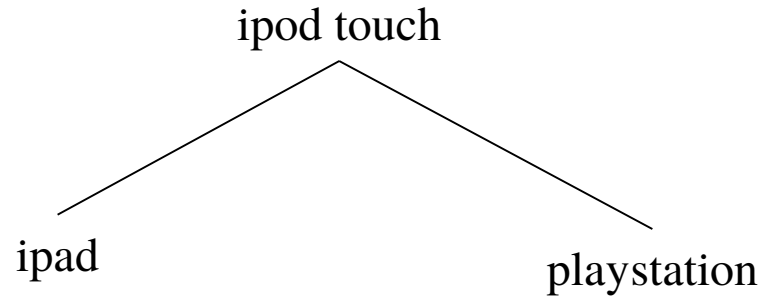
iphone

~~ipod touch~~

nintendo wii

~~playstation~~

smart tv



# Árboles

---

apple tv

~~ipad~~

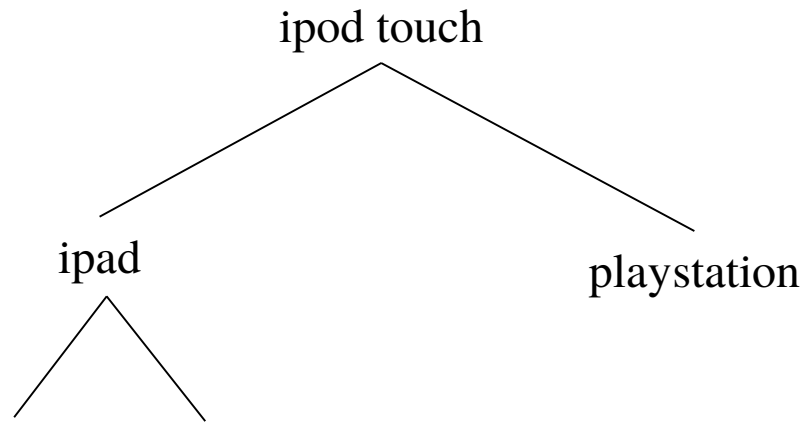
iphone

~~ipod touch~~

nintendo wii

~~playstation~~

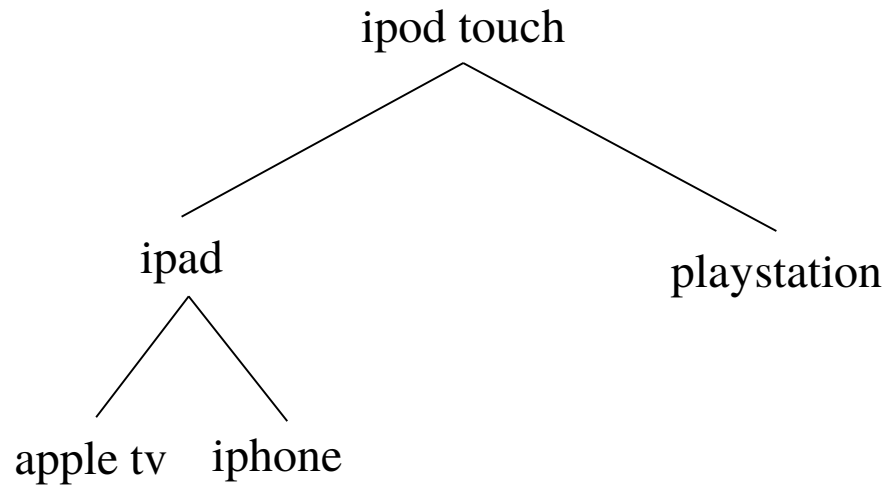
smart tv



# Árboles

---

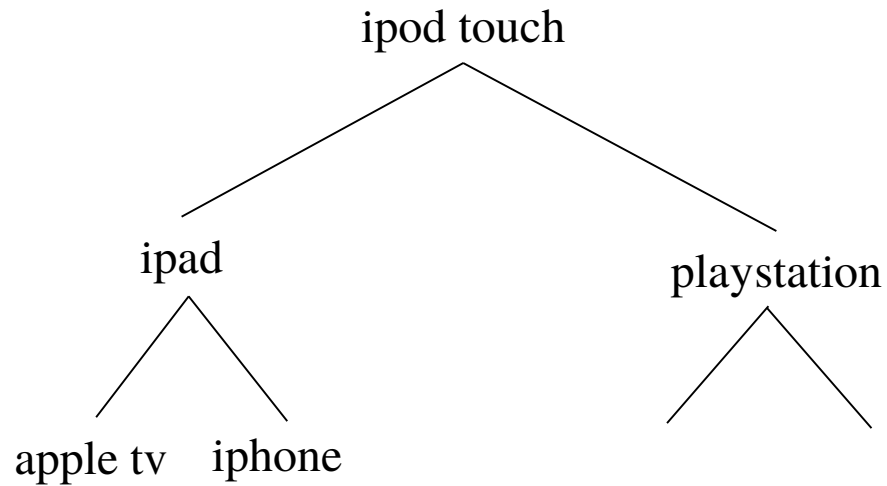
~~apple tv~~  
~~ipad~~  
~~iphone~~  
~~ipod touch~~  
nintendo wii  
~~playstation~~  
smart tv



# Árboles

---

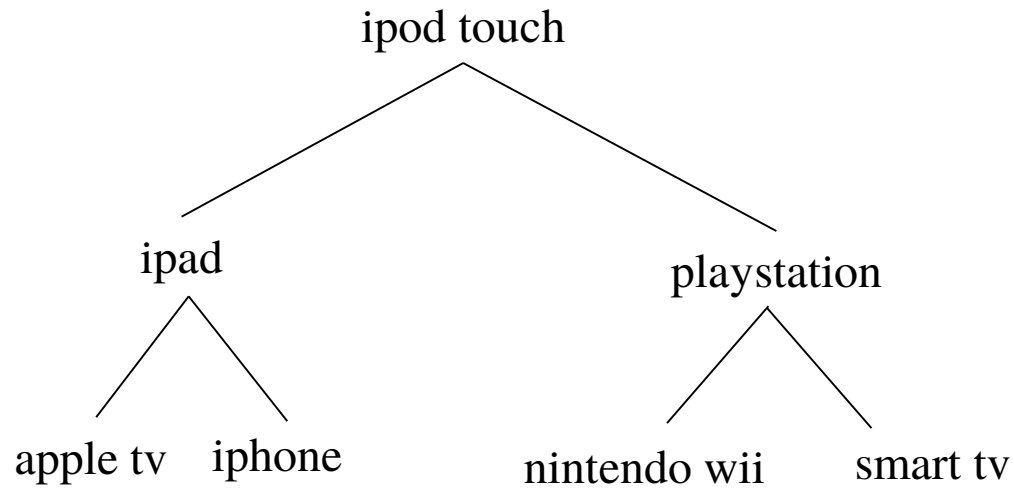
~~apple tv~~  
~~ipad~~  
~~iphone~~  
~~ipod touch~~  
nintendo wii  
~~playstation~~  
smart tv



# Árboles

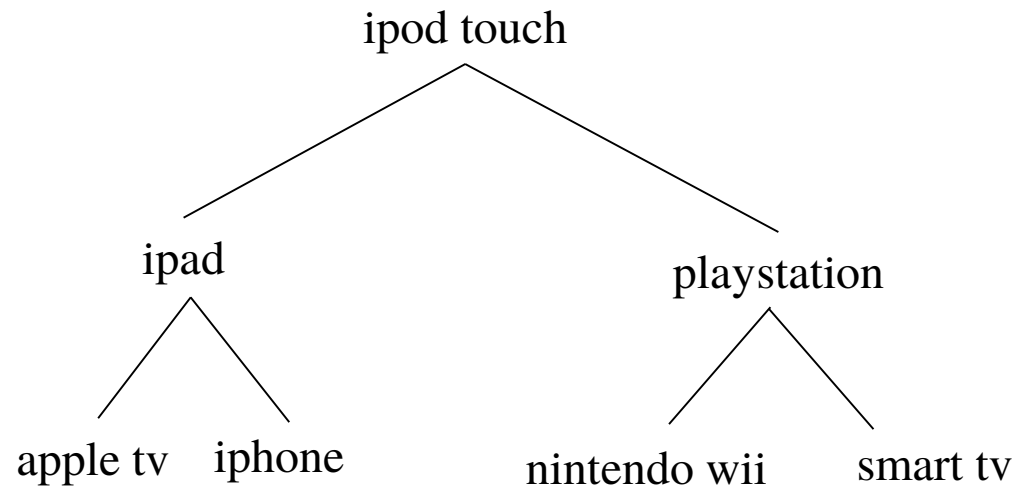
---

~~apple tv~~  
~~ipad~~  
~~iphone~~  
~~ipod touch~~  
~~nintendo wii~~  
~~playstation~~  
~~smart tv~~



# Árboles

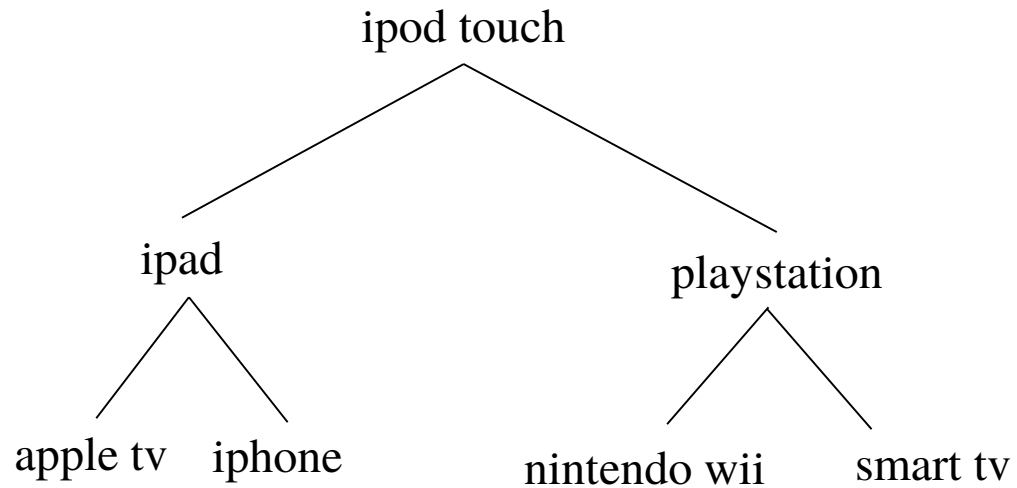
---



# Árboles

---

Insertar la llave kindle

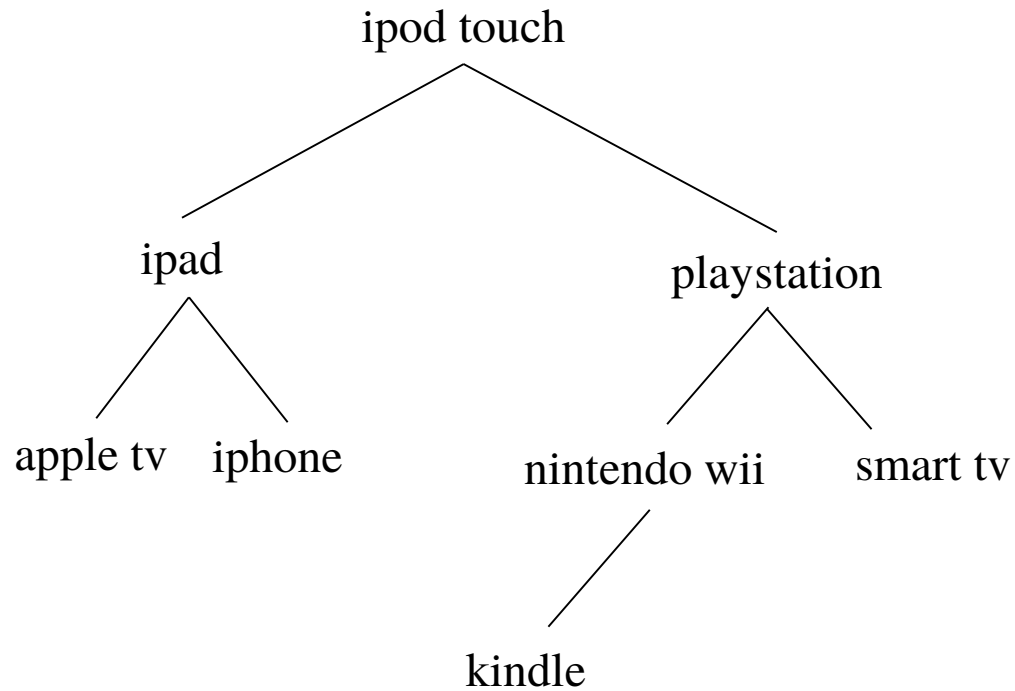




# Árboles

---

Insertar la llave kindle



# Árboles

---

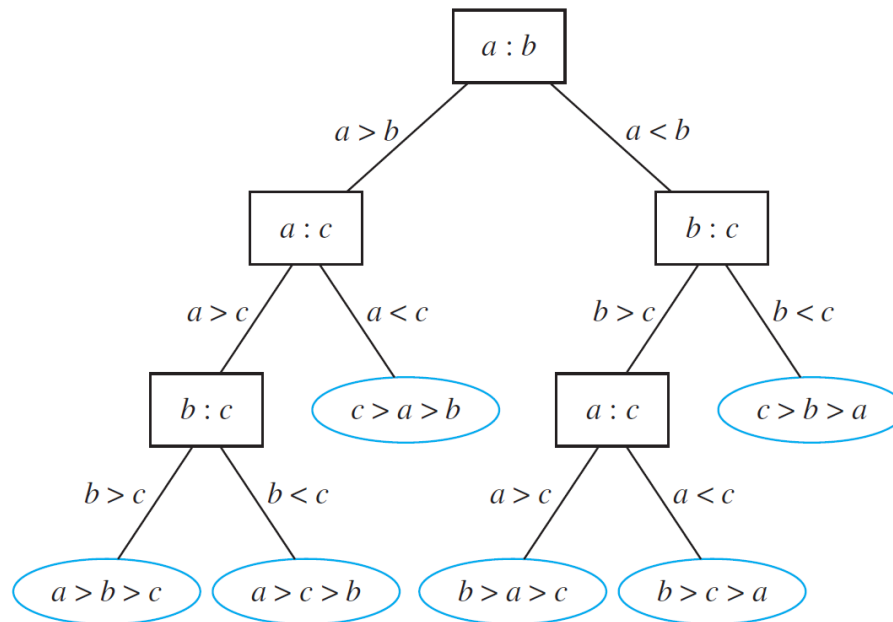
## Aplicaciones de los árboles

- Árboles de juego
- Árboles binarios de búsqueda
- Árboles de decisión

# Árboles

## Árboles de decisión

Es un árbol en cuyos vértices se tienen condiciones y en las hojas decisiones sobre un problema particular



Árbol de decisión para  
ordenar 3 elementos

# Árboles

---

## Recorridos de los árboles

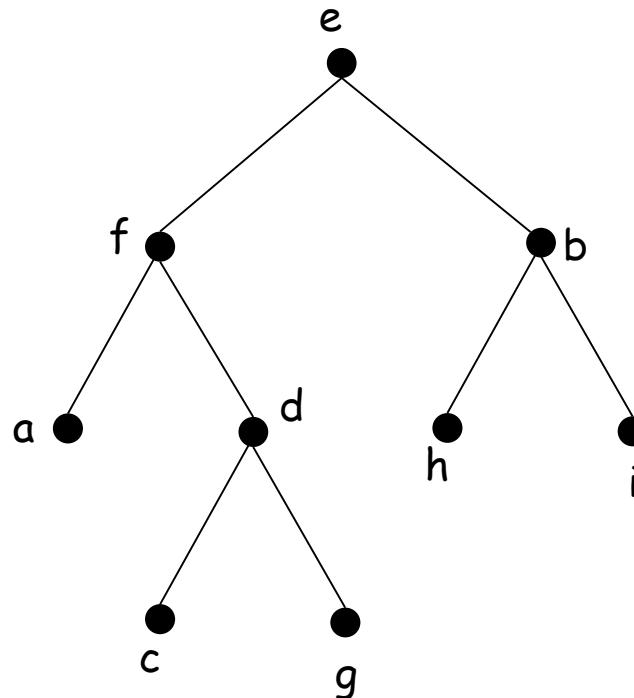
- Preorden
- Inorden
- Postorden

# Árboles

---

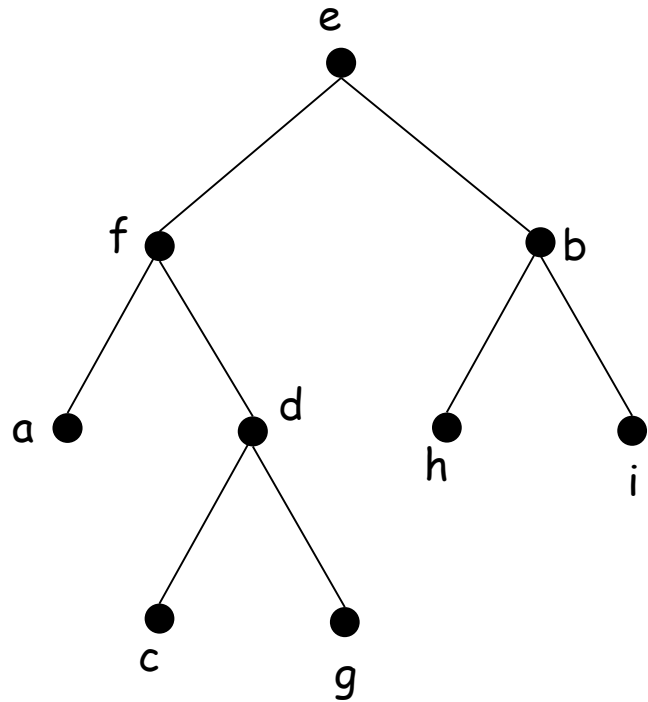
## Recorridos en preorden

Sea  $T$  un árbol con raíz  $r$  y subárboles  $T_1, T_2, \dots, T_n$ . El recorrido en preorden se hace visitando  $r$ , luego  $T_1$  en preorden,  $T_2$  en preorden, así hasta  $T_n$  en preorden



# Árboles

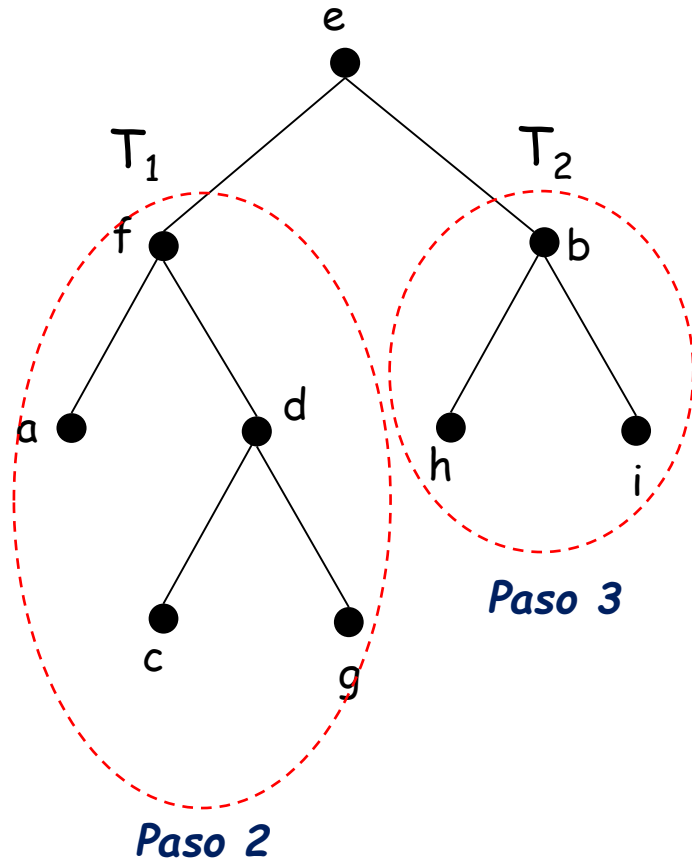
---



Recorrido en preorden:

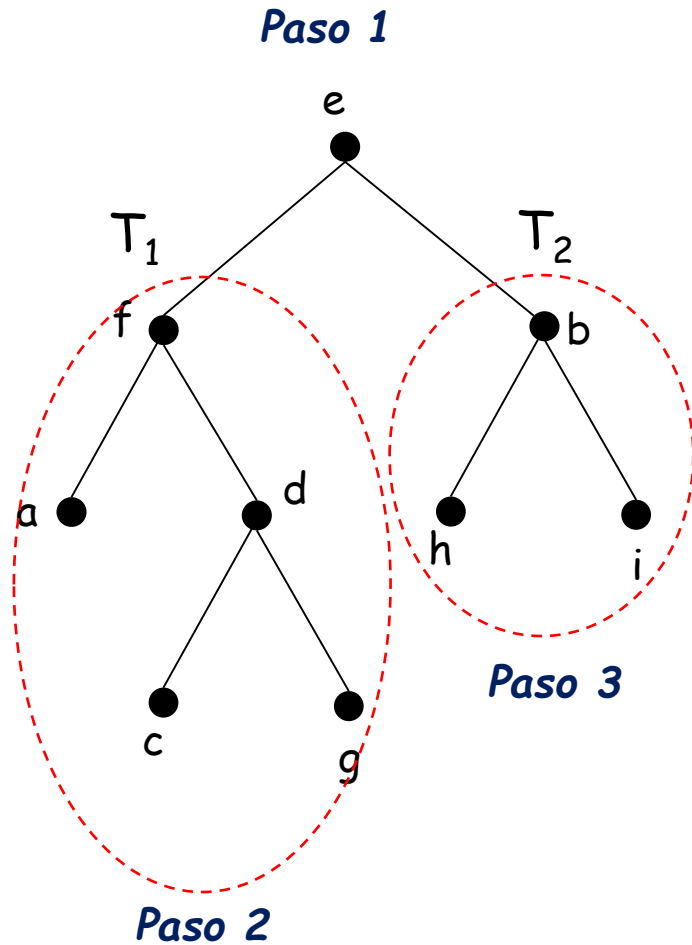
# Árboles

*Paso 1*



Recorrido en preorden:

# Árboles



Recorrido en preorden:

e

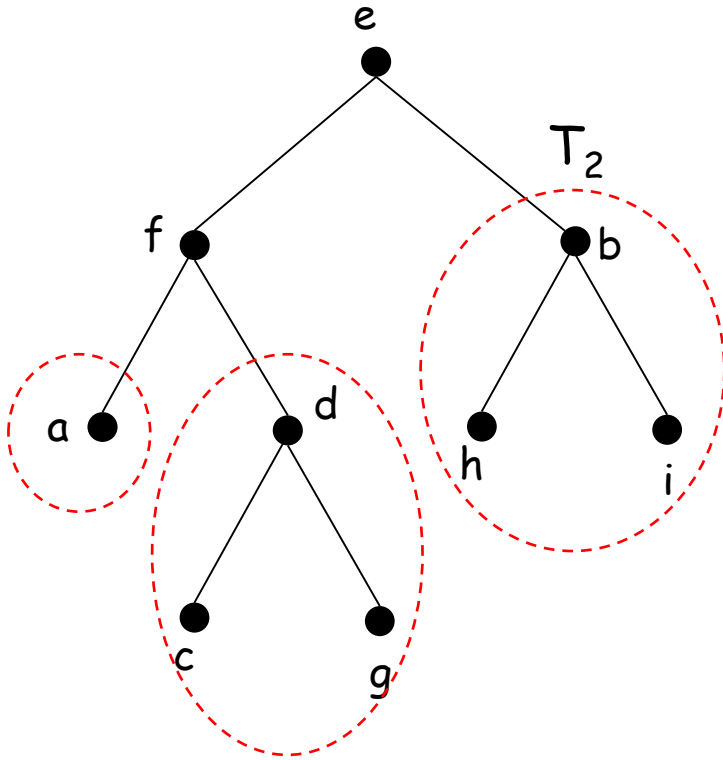


# Árboles

---

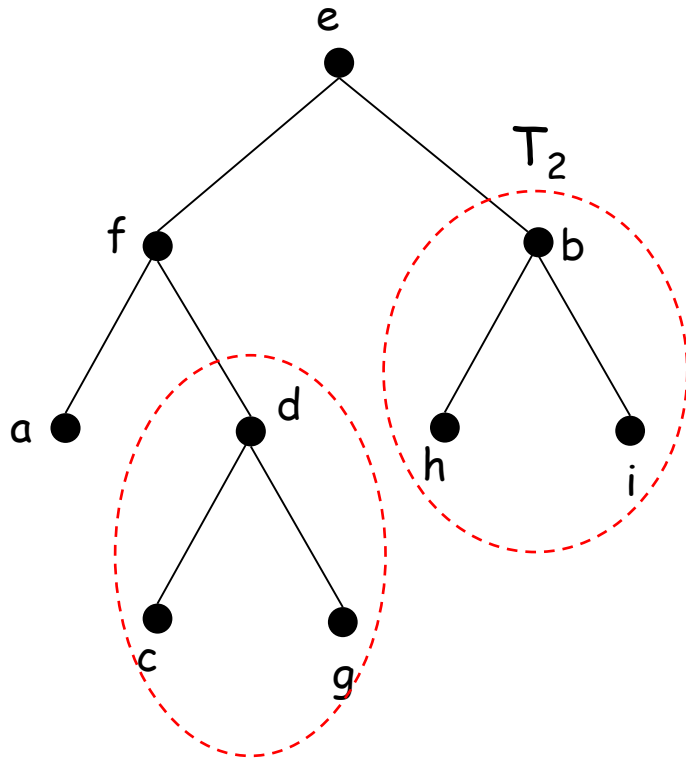
Recorrido en preorden:

$e - f$



# Árboles

---

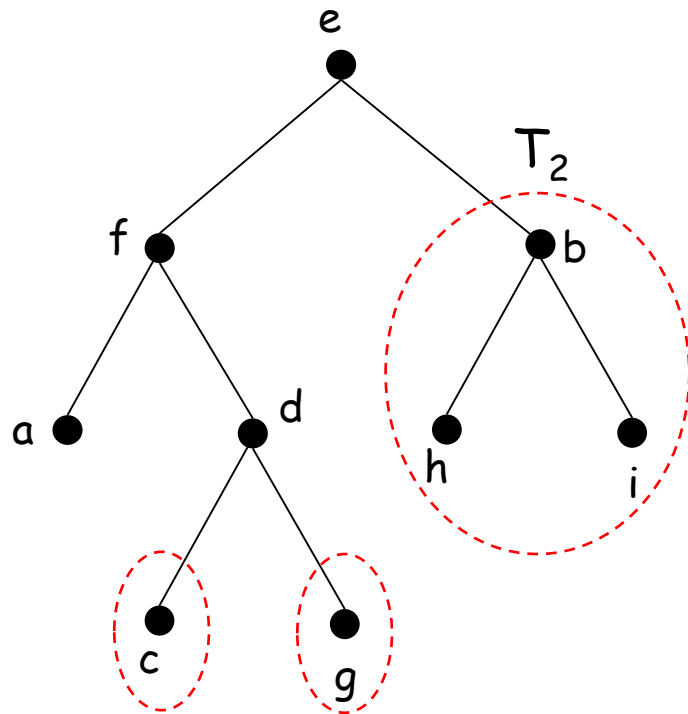


Recorrido en preorden:

e - f - a

# Árboles

---

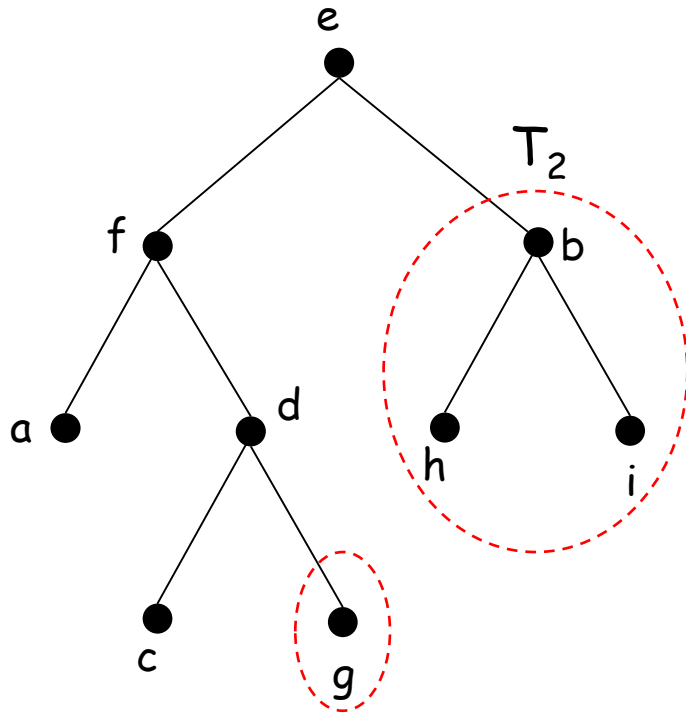


Recorrido en preorden:

$e - f - a - d$

# Árboles

---

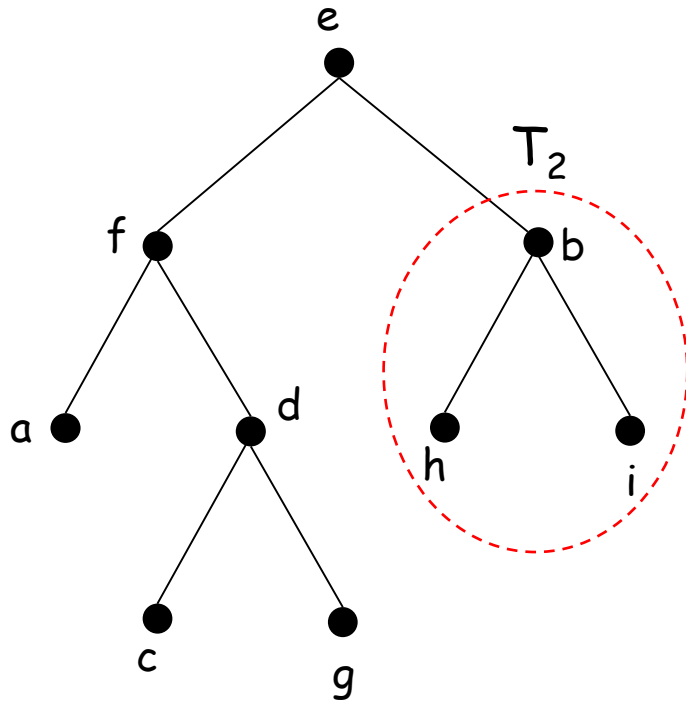


Recorrido en preorden:

$e - f - a - d - c$

# Árboles

---

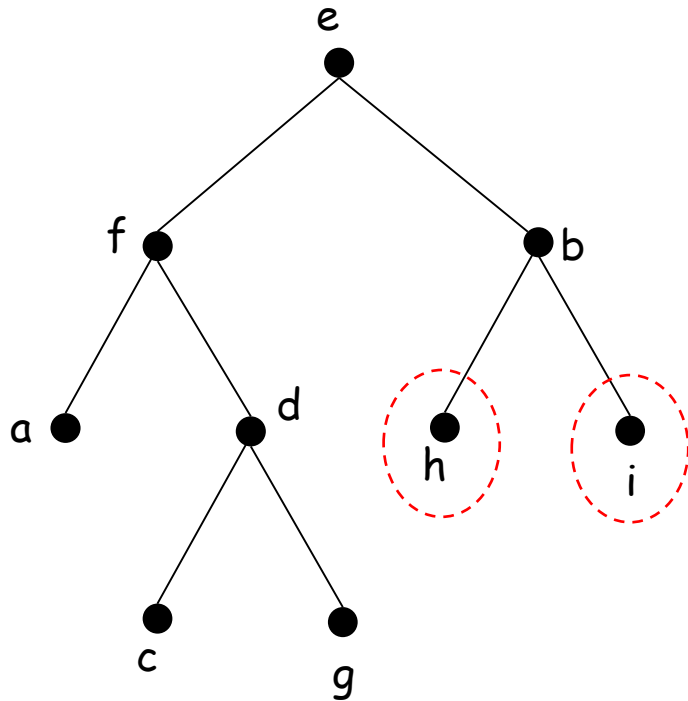


Recorrido en preorden:

$e - f - a - d - c - g$

# Árboles

---

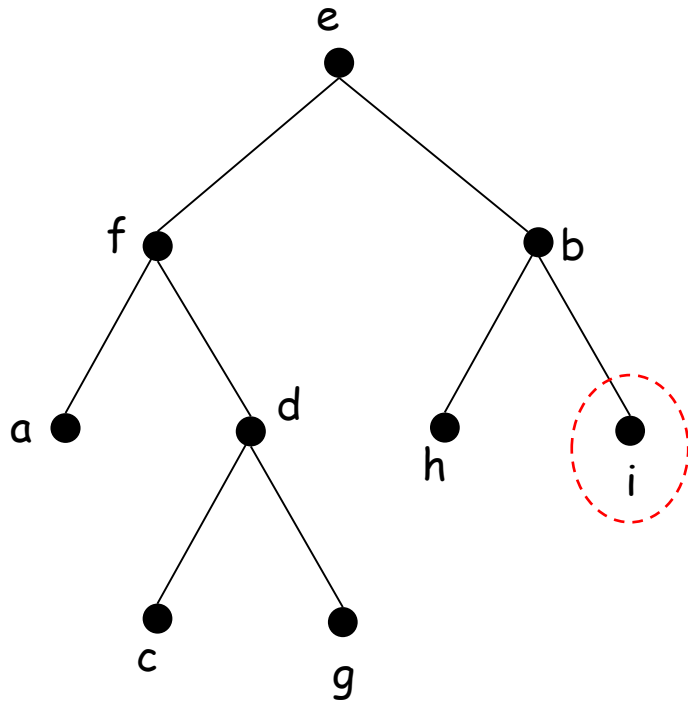


Recorrido en preorden:

$e - f - a - d - c - g - b$

# Árboles

---

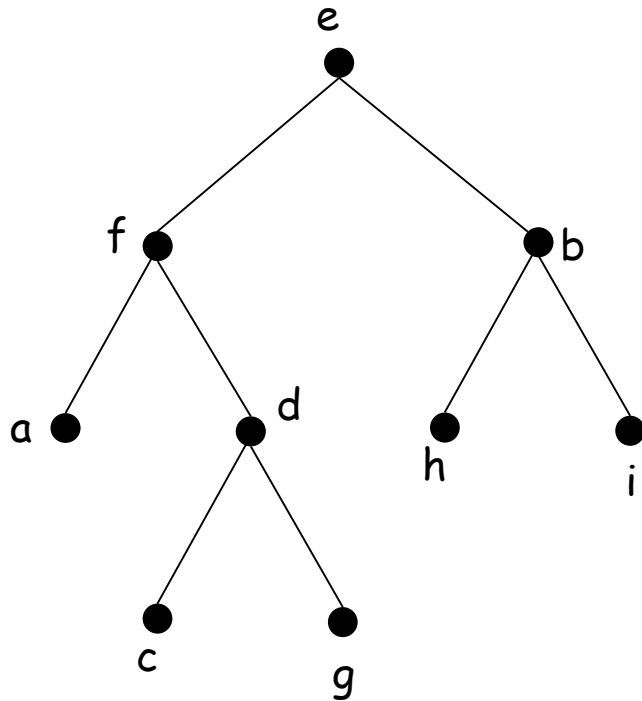


Recorrido en preorden:

$e - f - a - d - c - g - b - h$

# Árboles

---



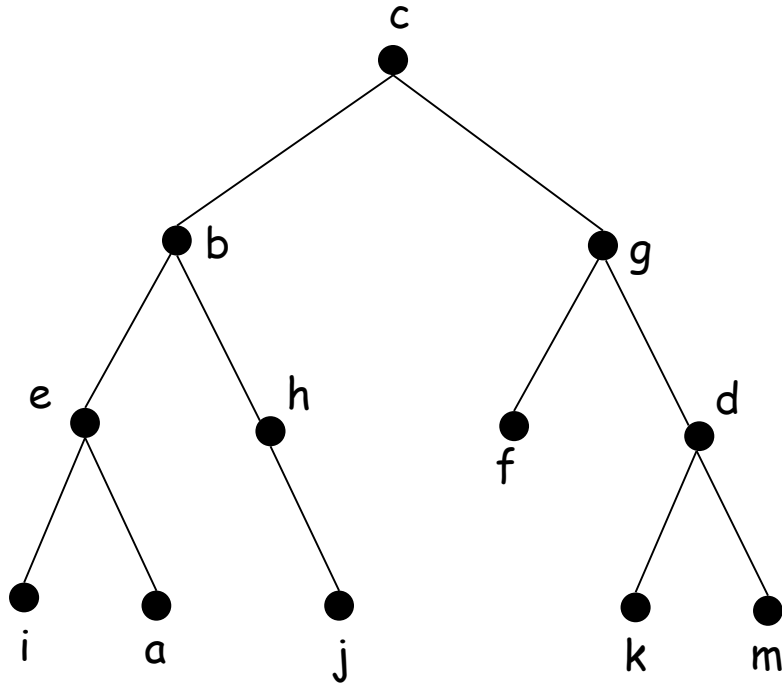
Recorrido en preorden:

$e - f - a - d - c - g - b - h - i$



# Árboles

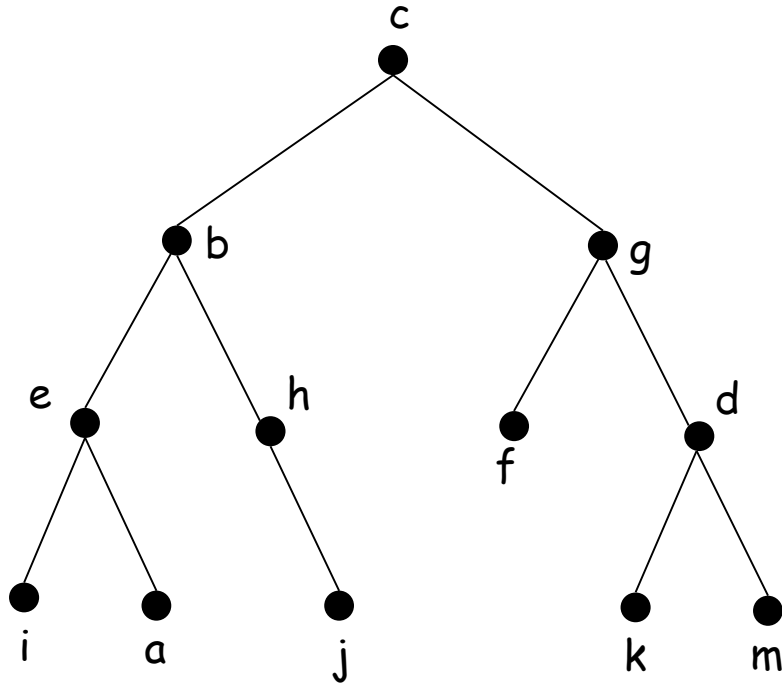
---



Recorrido en preorden:

# Árboles

---

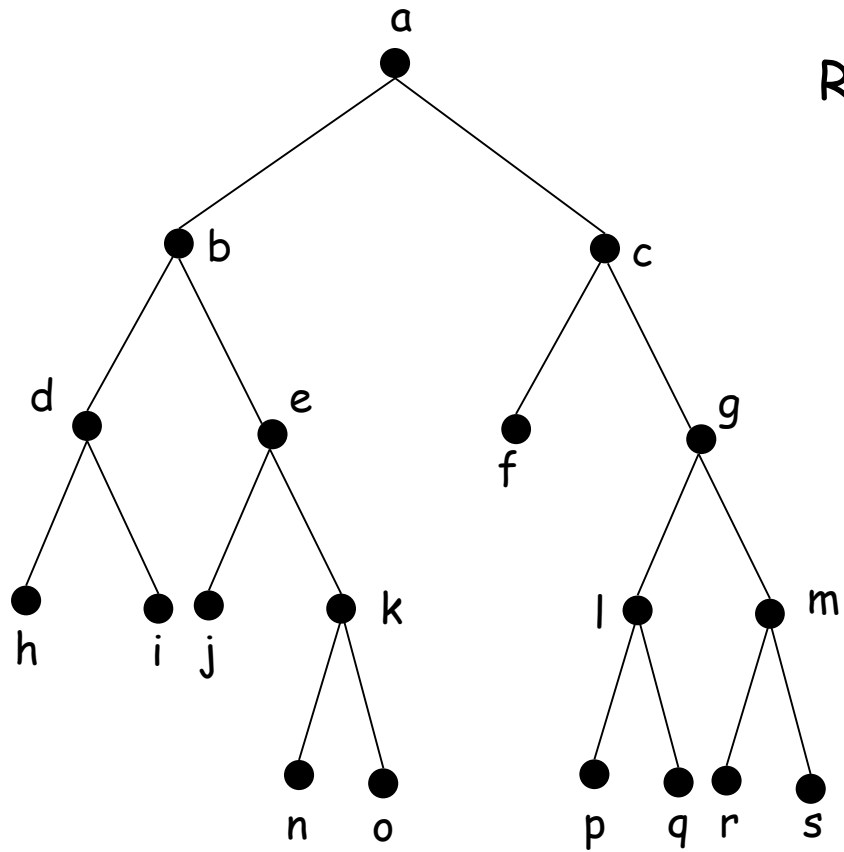


Recorrido en preorden:

c - b - e - i - a - h - j - g - f - d - k - m

# Árboles

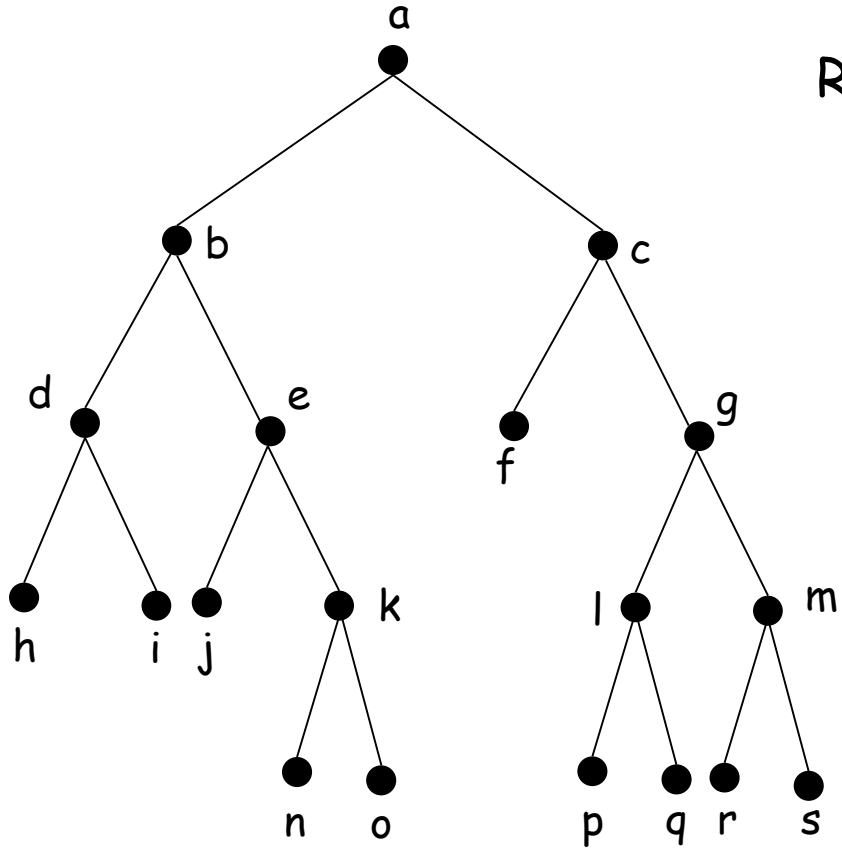
---



Recorrido en preorden:

# Árboles

---



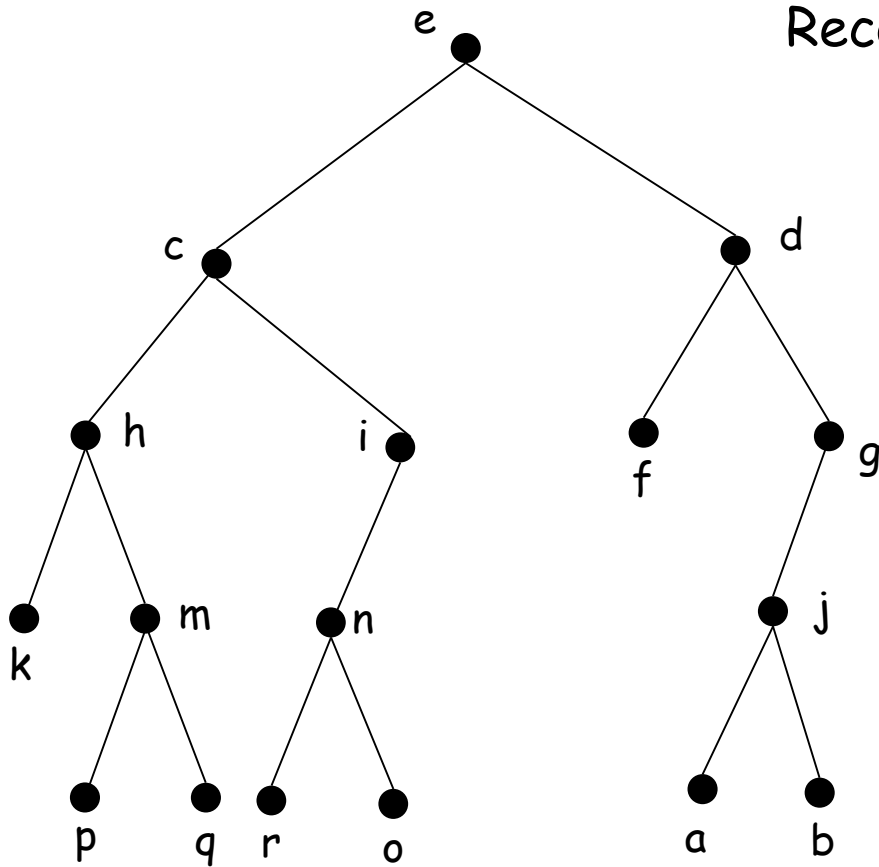
Recorrido en preorden:

a-b-d-h-i-e-j-k-n-o-c-f-g-l-p-q-m-r-s

# Árboles

---

Recorrido en preorden:

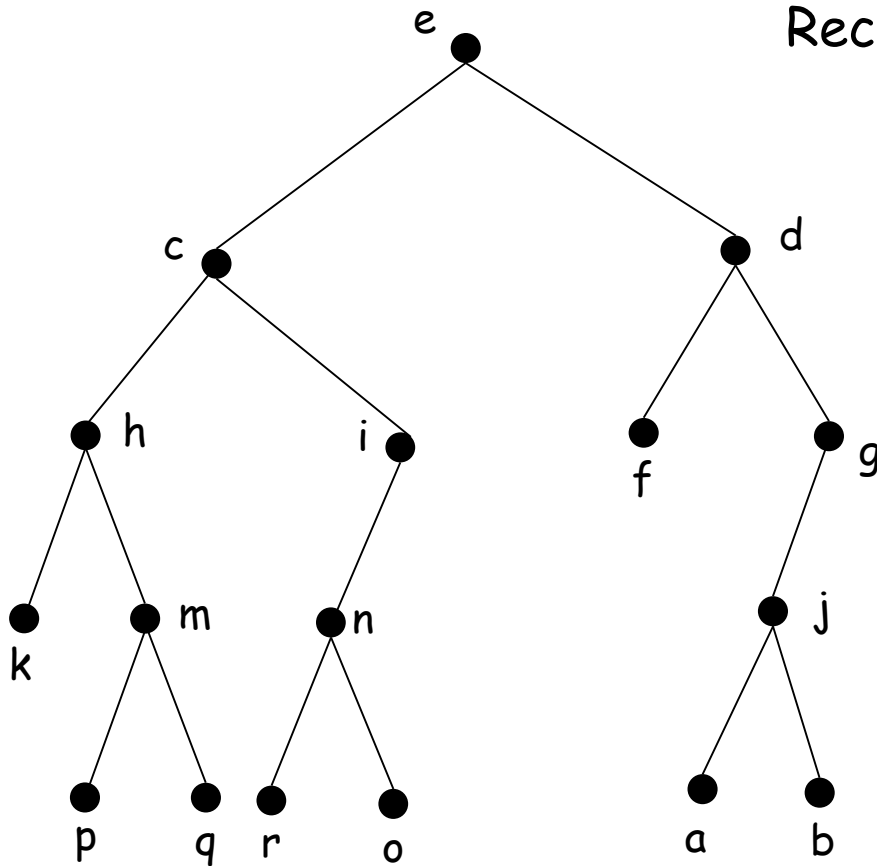


# Árboles

---

Recorrido en preorden:

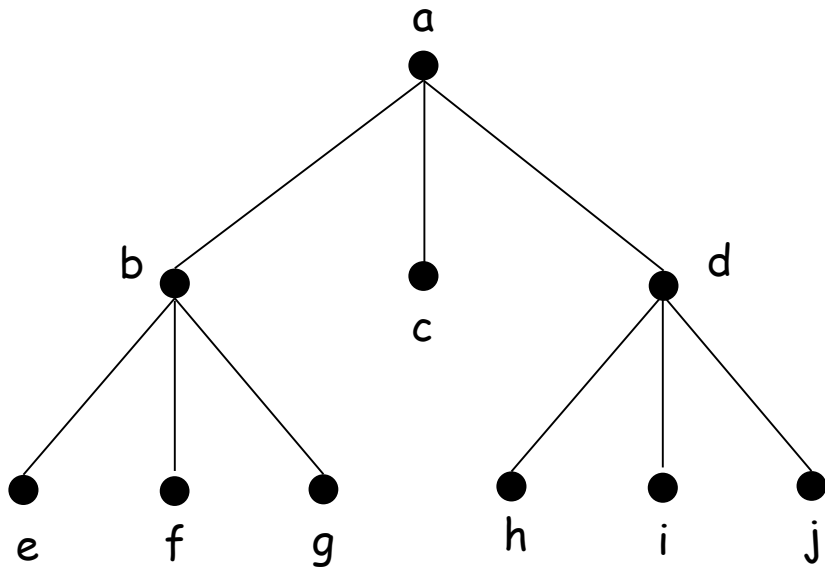
e-c-h-k-m-p-q-i-n-r-o-d-f-g-j-a-b



# Árboles

---

Recorrido en preorden:

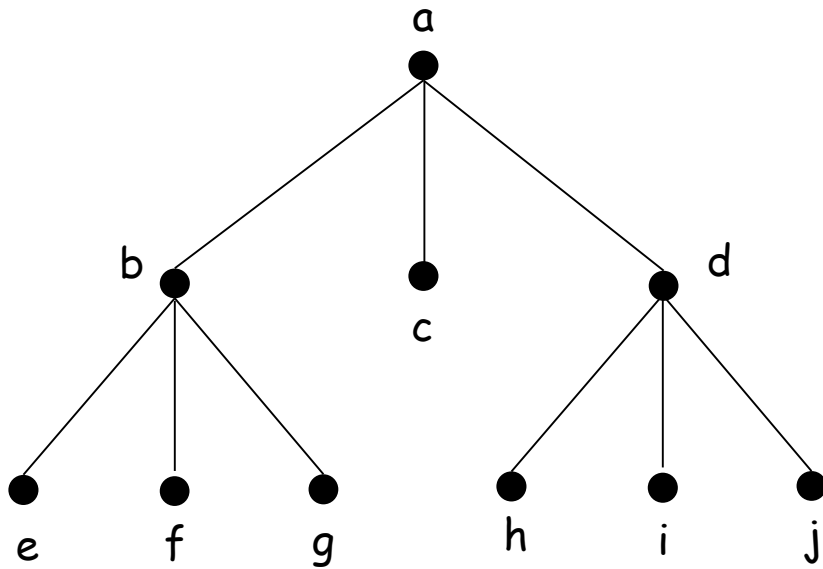


# Árboles

---

Recorrido en preorden:

a - b - e - f - g - c - d - h - i - j

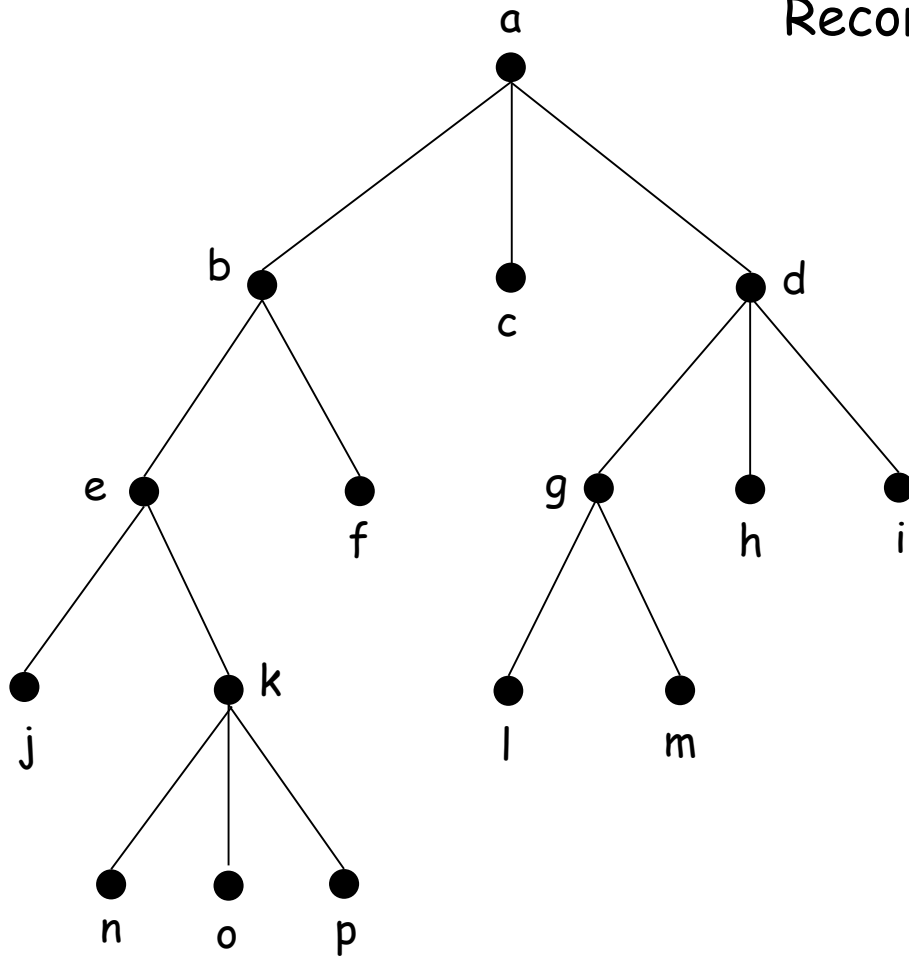




# Árboles

---

Recorrido en preorden:

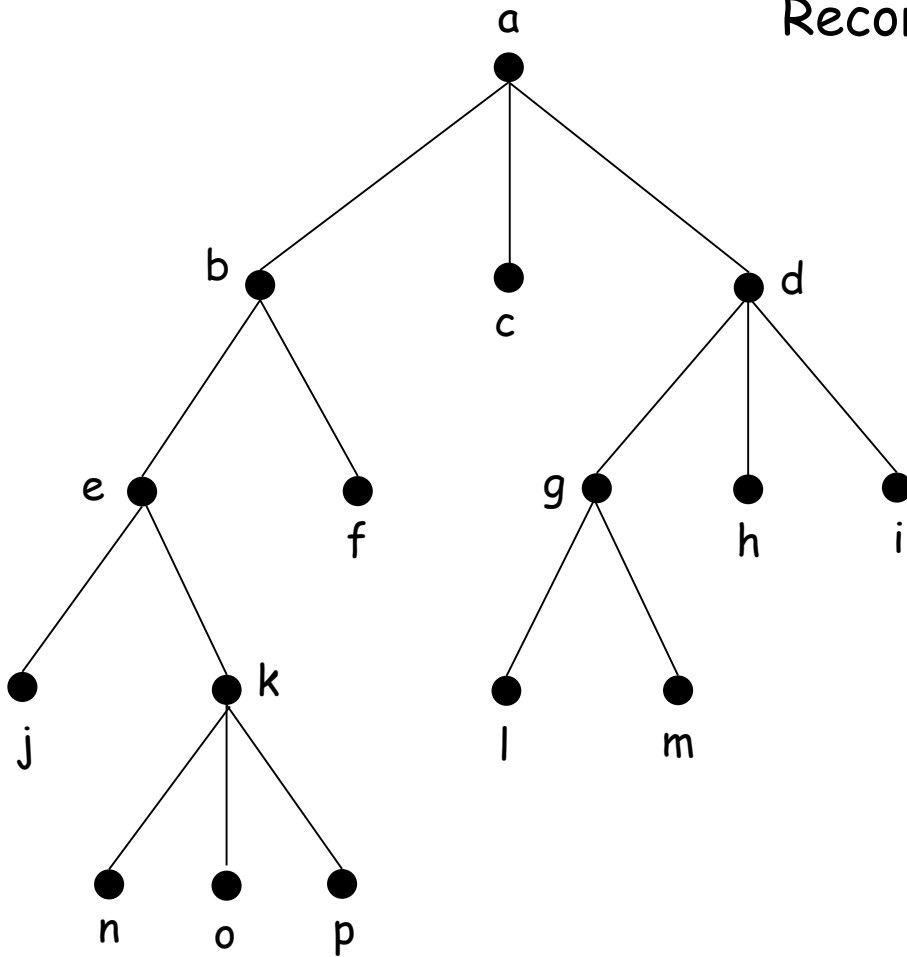


# Árboles

---

Recorrido en preorden:

a-b-e-j-k-n-o-p-f-c-d-g-l-m-h-i



# Árboles

---

## Recorridos de los árboles

- Preorden
- Inorden
- Postorden

# Árboles

---

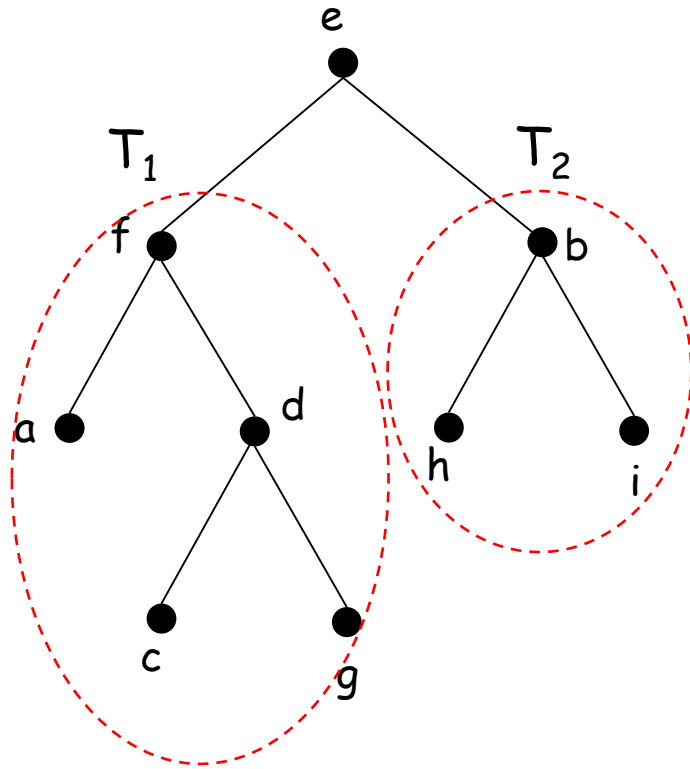
## Recorridos en inorden

Sea  $T$  un árbol con raíz  $r$  y subárboles  $T_1, T_2, \dots, T_n$ . El recorrido en inorden se hace realizando el recorrido de  $T_1$  en inorden, luego visitando  $r$ , luego se hace el recorrido de  $T_2$  en inorden, así hasta  $T_n$  en inorden

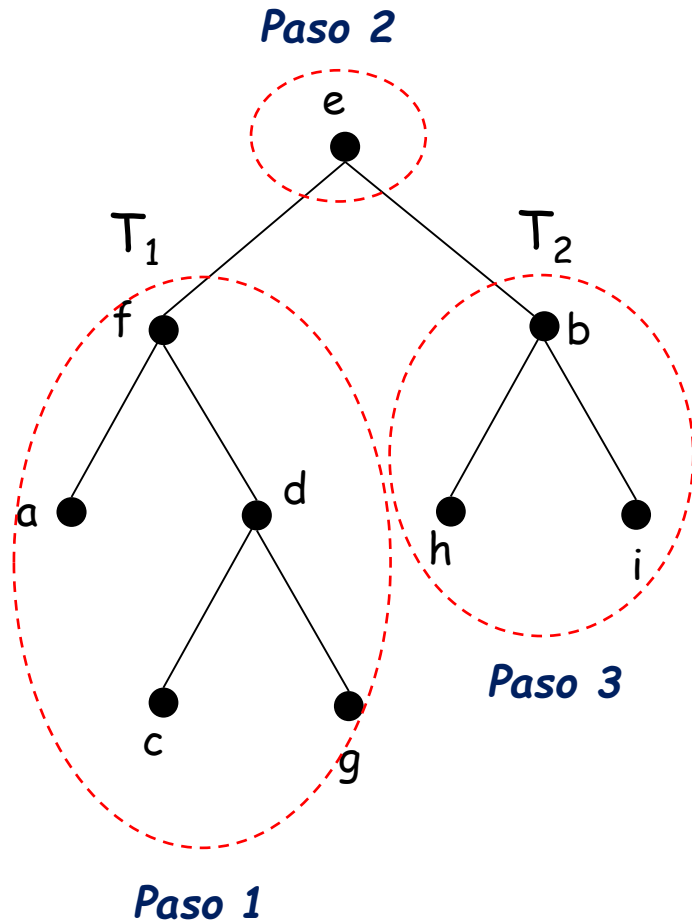
# Árboles

---

Recorrido en inorden:



# Árboles

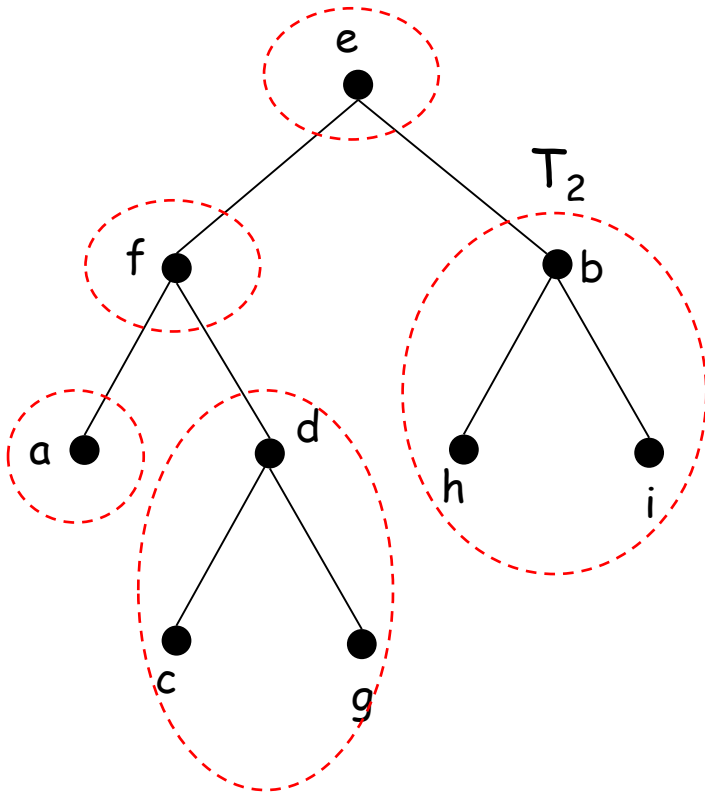


Recorrido en inorden:

# Árboles

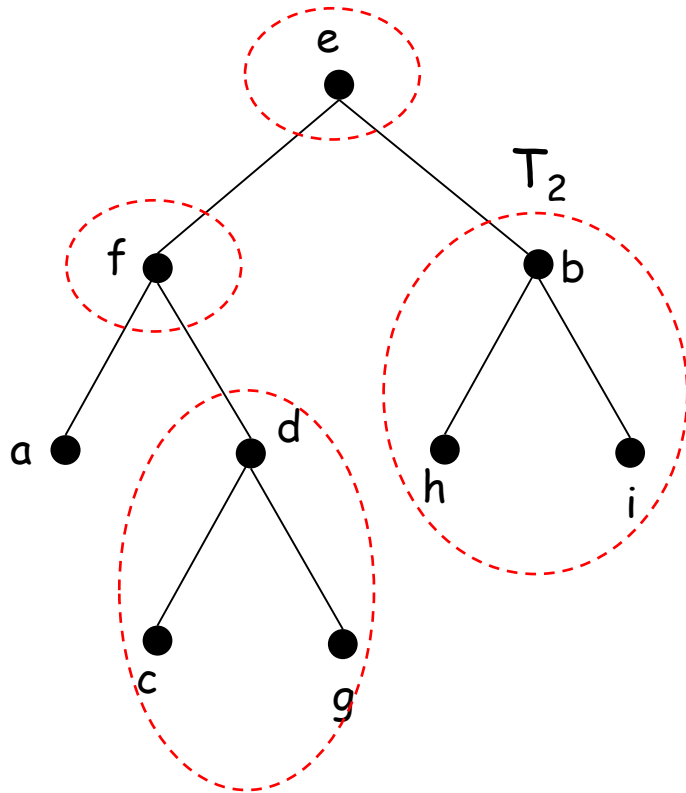
---

Recorrido en inorden:



# Árboles

---



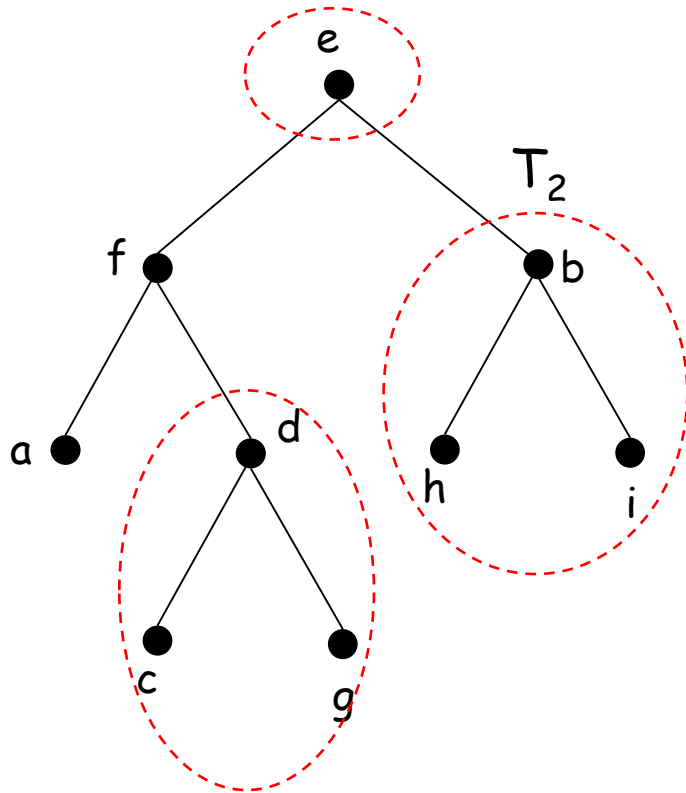
Recorrido en inorden:

a



# Árboles

---

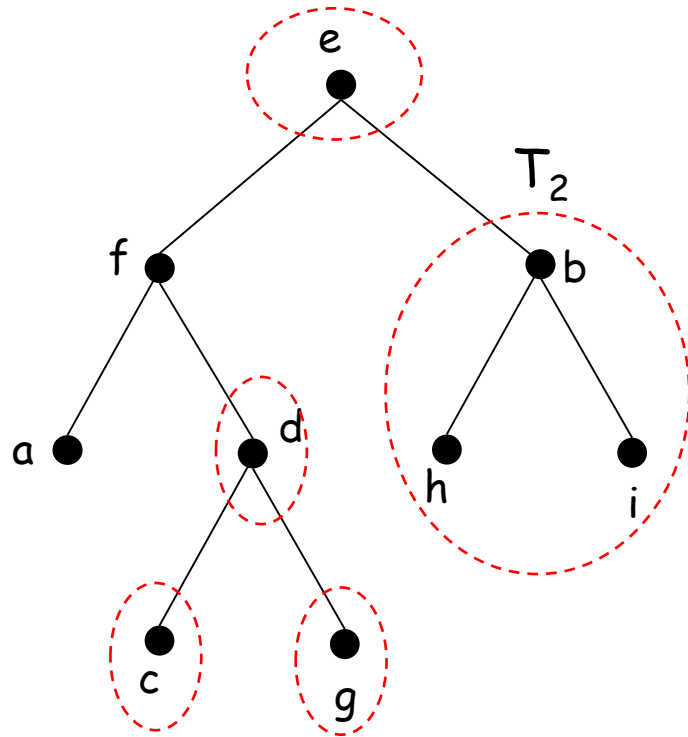


Recorrido en inorden:

a - f

# Árboles

---

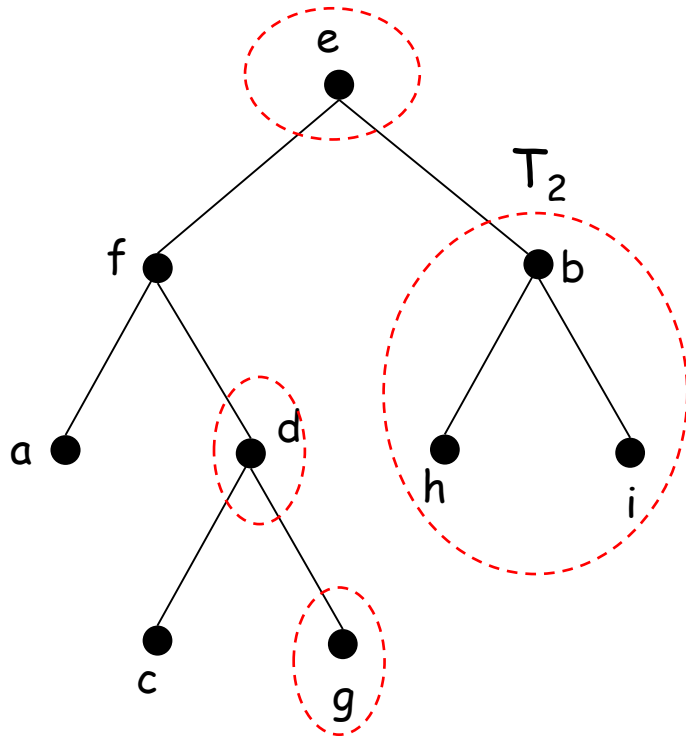


Recorrido en inorden:

a - f

# Árboles

---

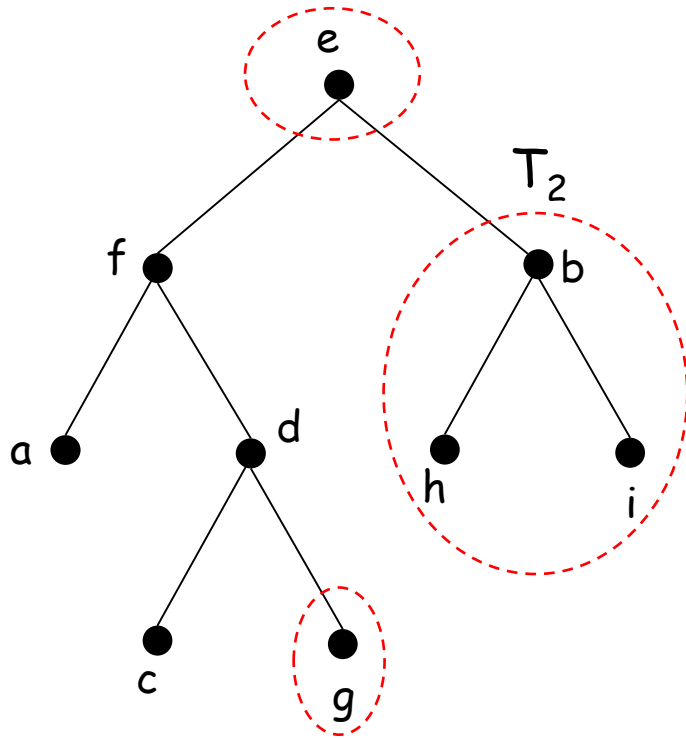


Recorrido en inorden:

a - f - c

# Árboles

---

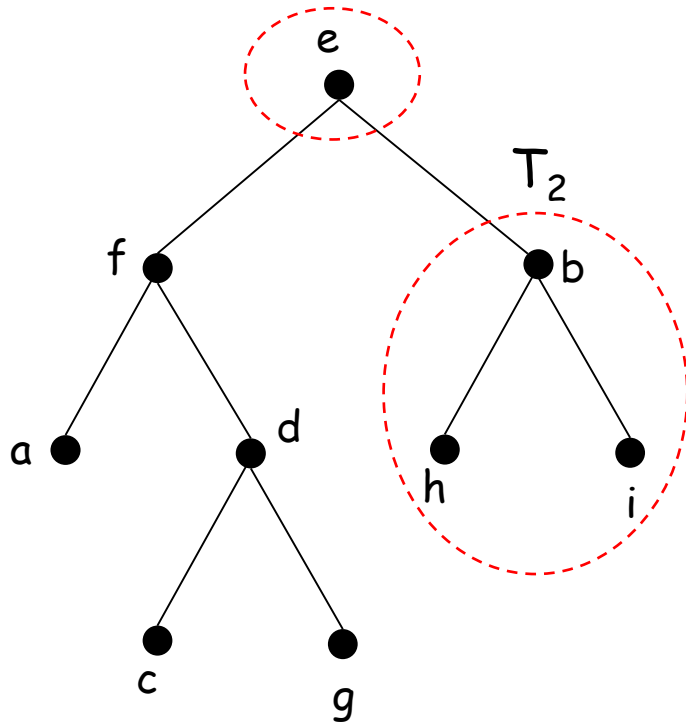


Recorrido en inorden:

a - f - c - d

# Árboles

---

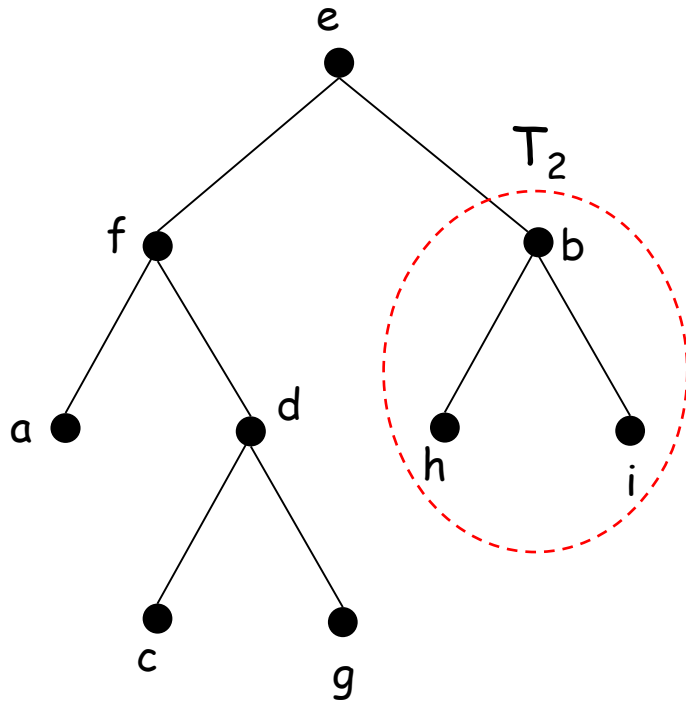


Recorrido en inorden:

a - f - c - d - g

# Árboles

---

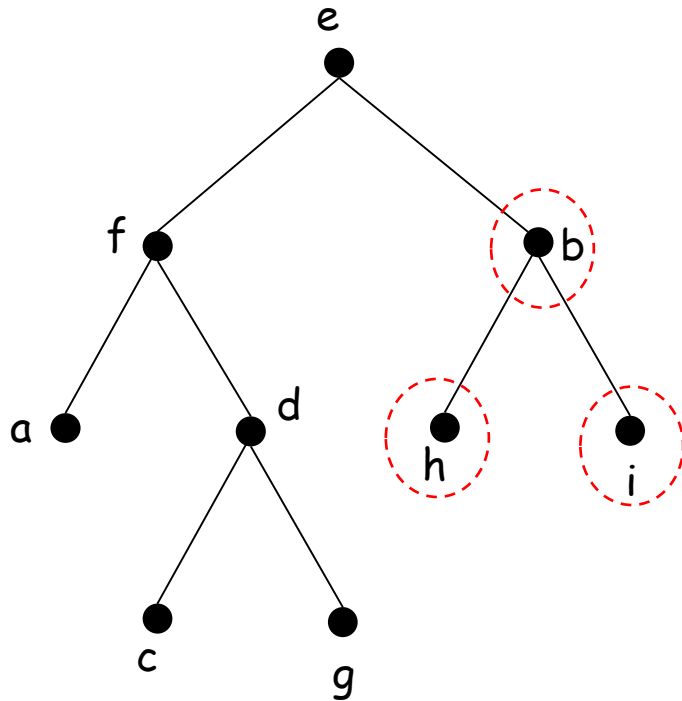


Recorrido en inorden:

a - f - c - d - g - e

# Árboles

---

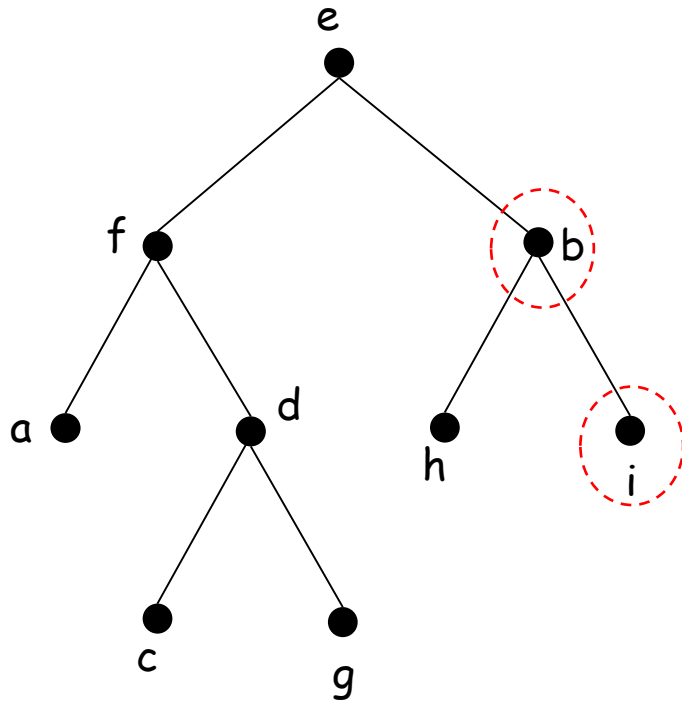


Recorrido en inorden:

a - f - c - d - g - e

# Árboles

---



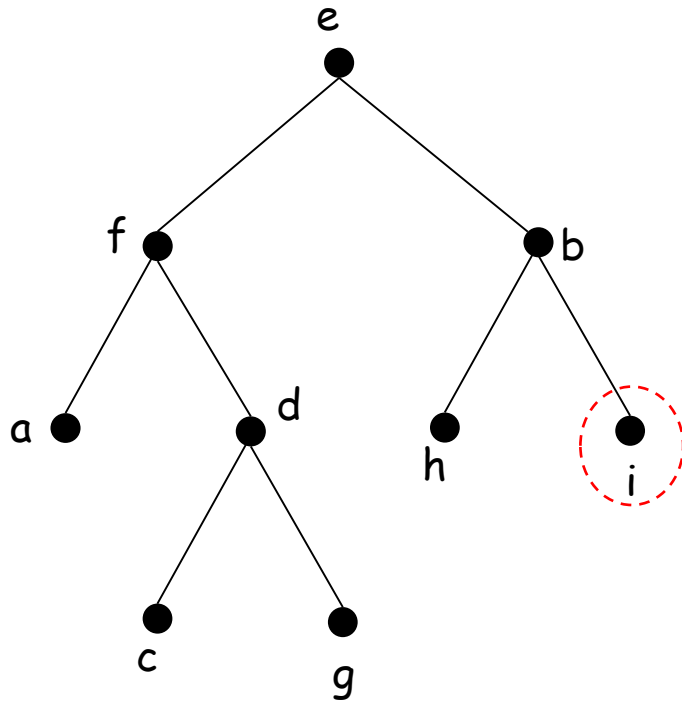
Recorrido en inorden:

a - f - c - d - g - e - h



# Árboles

---

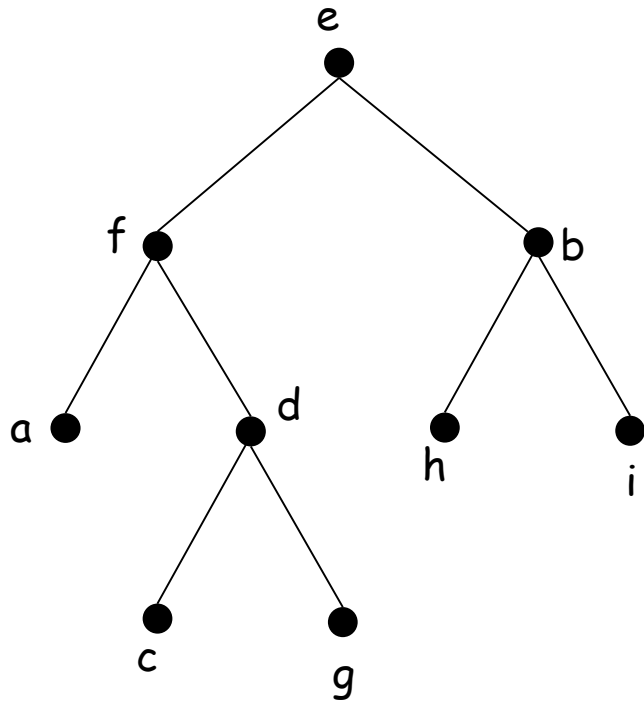


Recorrido en inorden:

a - f - c - d - g - e - h - b

# Árboles

---

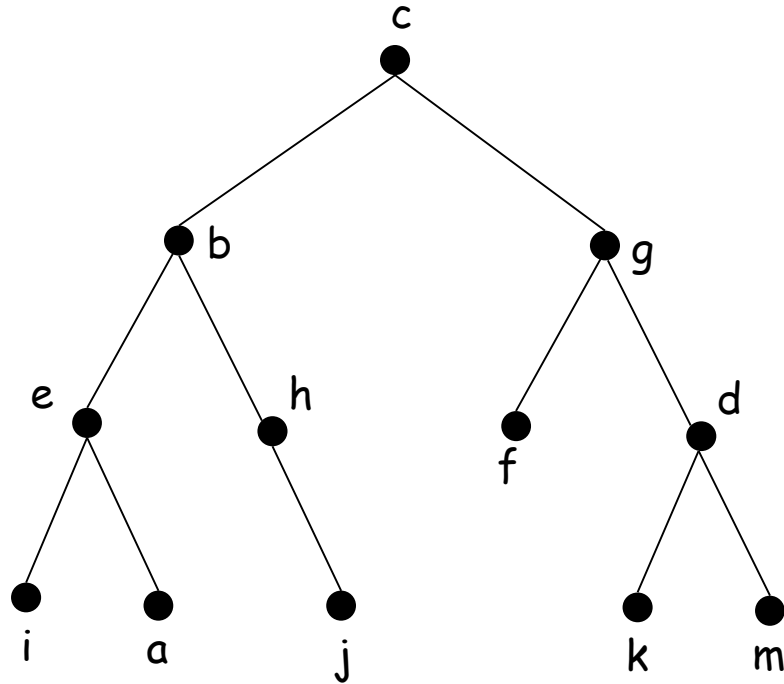


Recorrido en inorden:

a - f - c - d - g - e - h - b - i

# Árboles

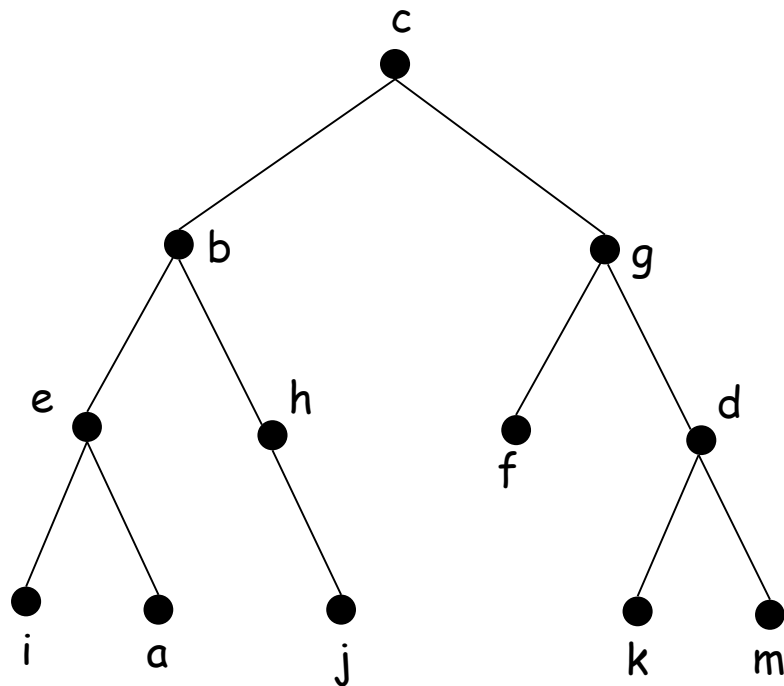
---



Recorrido en inorden:

# Árboles

---

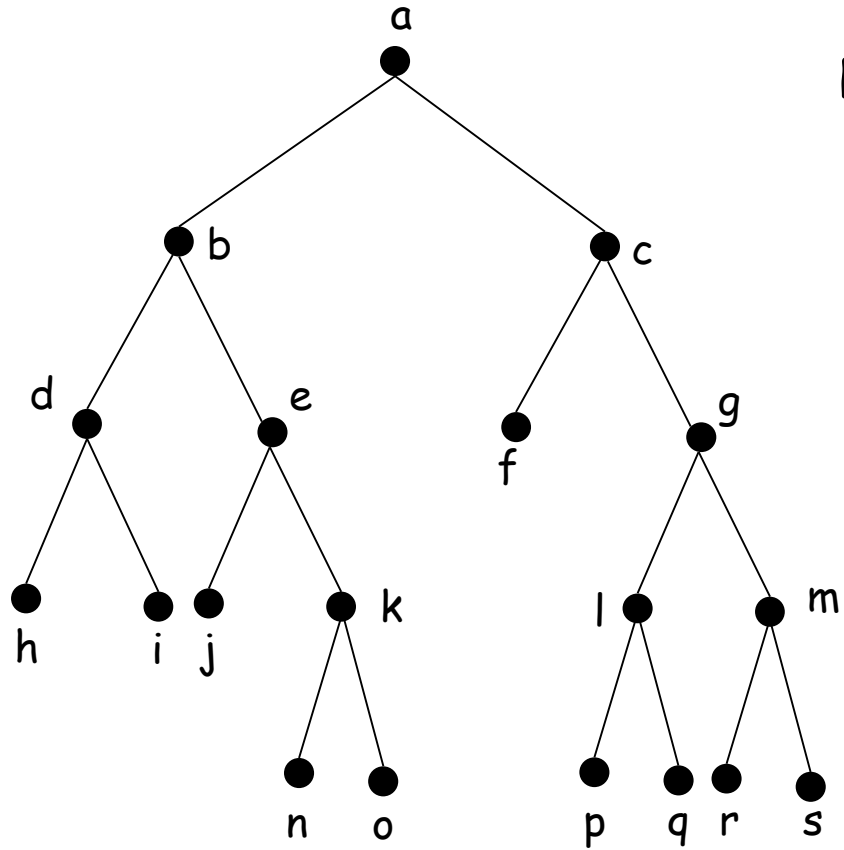


Recorrido en inorden:

i - e - a - b - h - j - c - f - g - k - d - m

# Árboles

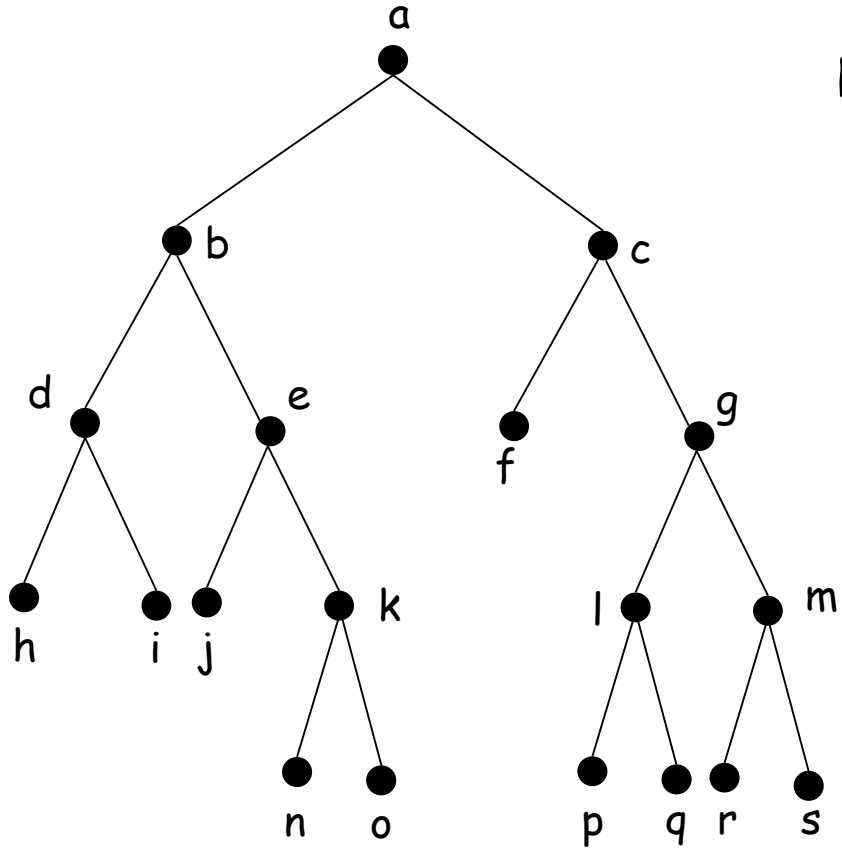
---



Recorrido en inorden:

# Árboles

---



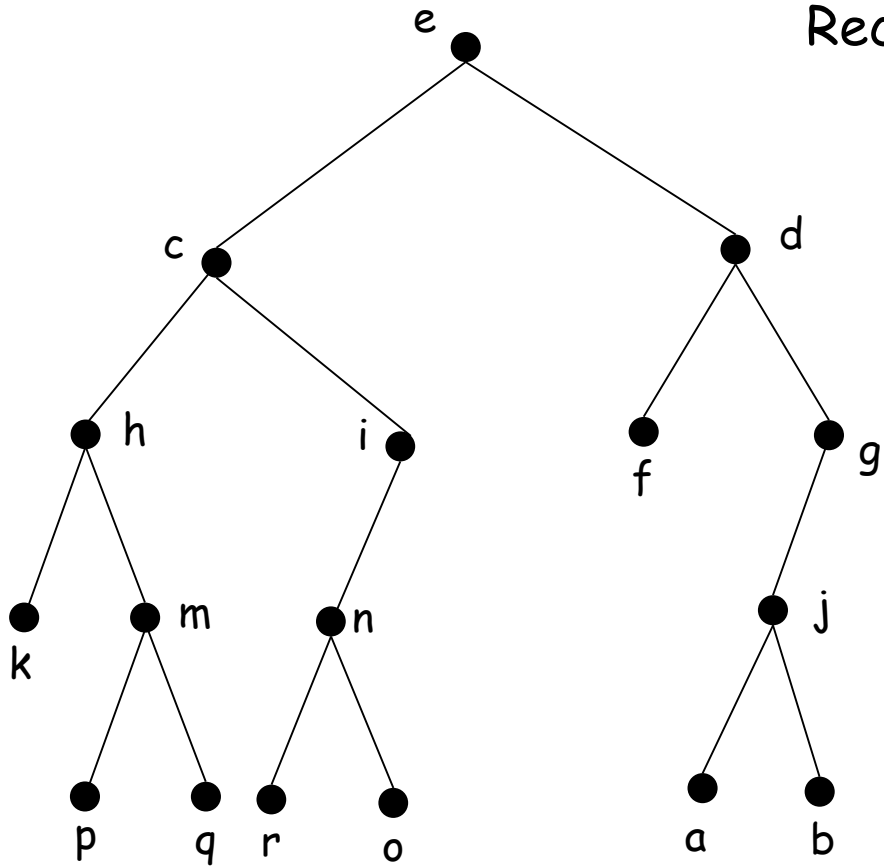
Recorrido en inorden:

h-d-i-b-j-e-n-k-o-a-f-c-p-l-q-g-r-m-s

# Árboles

---

Recorrido en inorden:

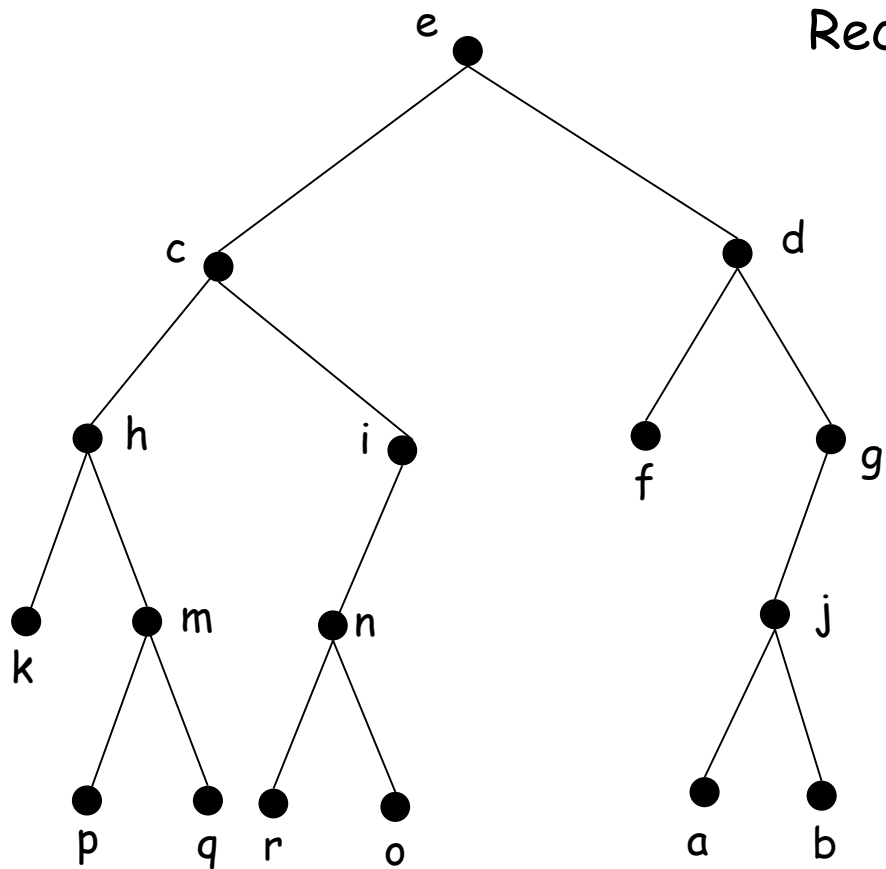


# Árboles

---

Recorrido en inorden:

k-h-p-m-q-c-r-n-o-i-e-f-d-a-j-b-g

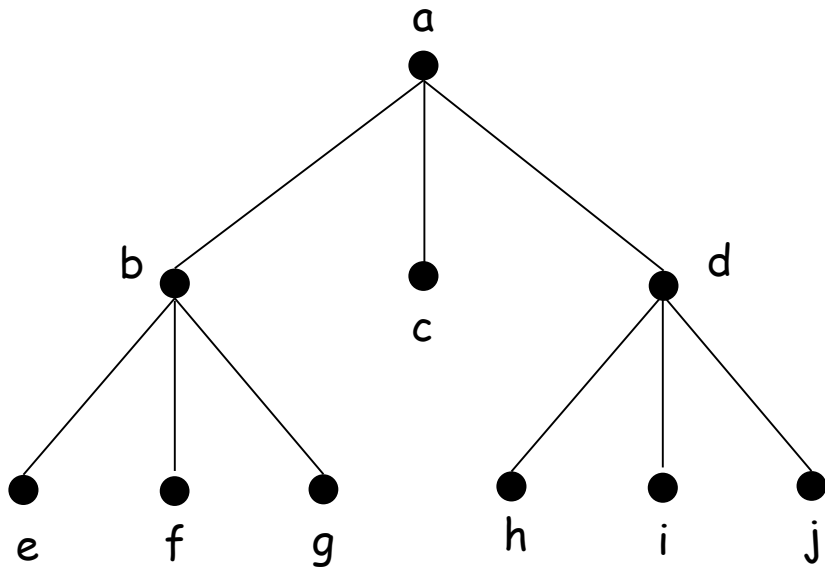




# Árboles

---

Recorrido en inorden:

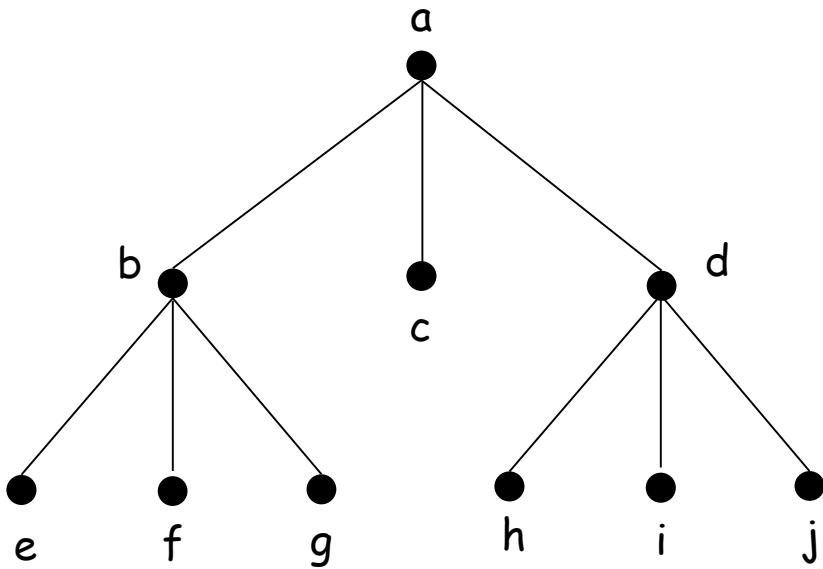


# Árboles

---

Recorrido en inorden:

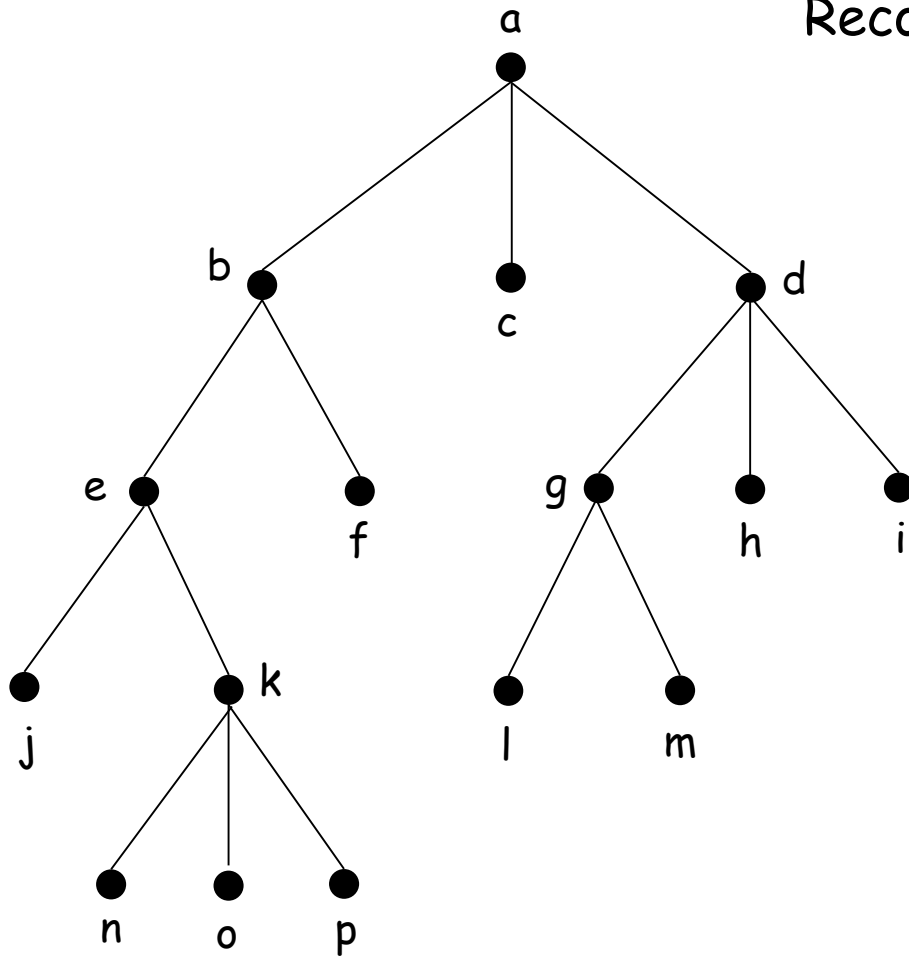
$e - b - f - g - a - c - h - d - i - j$



# Árboles

---

Recorrido en inorden:

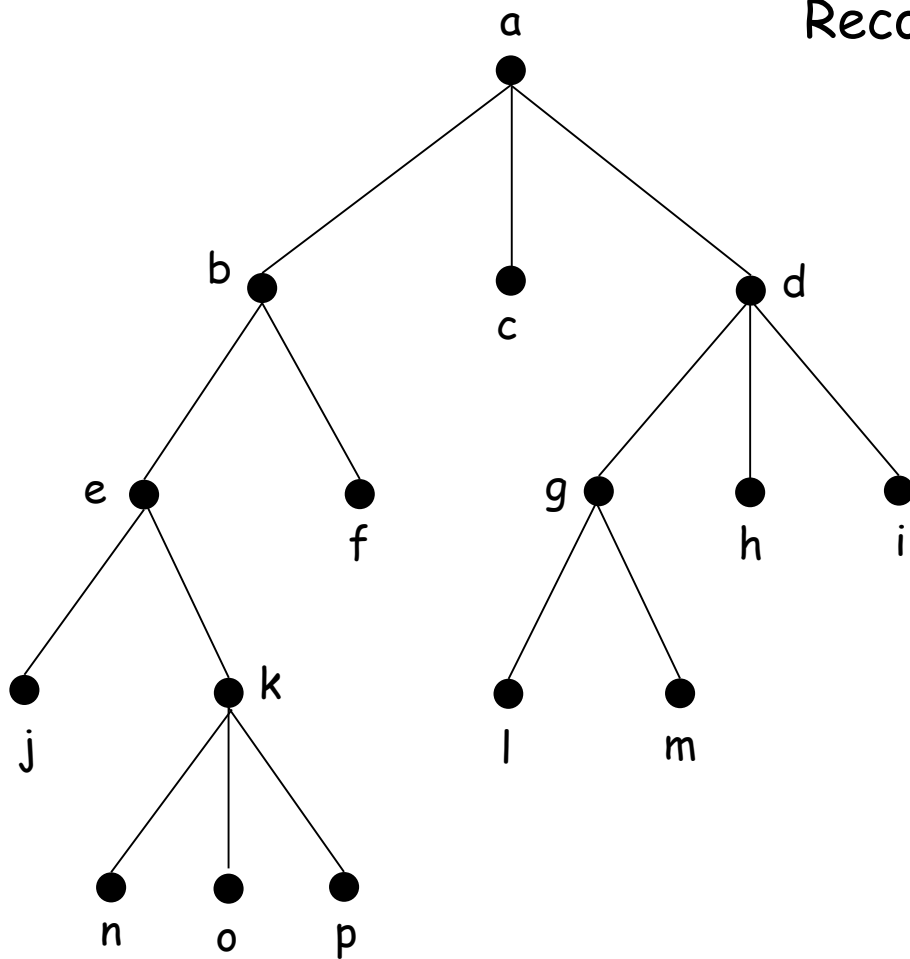


# Árboles

---

Recorrido en inorden:

j-e-n-k-o-p-b-f-a-c-l-g-m-d-h-i



# Árboles

---

## Recorridos de los árboles

- Preorden
- Inorden
- Postorden

# Árboles

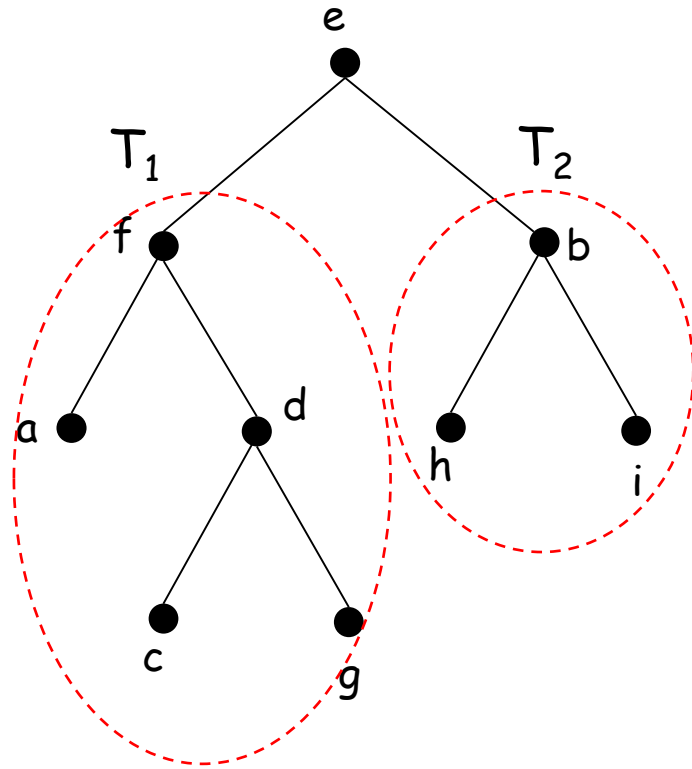
---

## Recorridos en postorden

Sea  $T$  un árbol con raíz  $r$  y subárboles  $T_1, T_2, \dots, T_n$ . El recorrido en postorden se hace realizando el recorrido de  $T_1$  en postorden,  $T_2$  en postorden, hasta  $T_n$  en postorden, y luego visitando  $r$

# Árboles

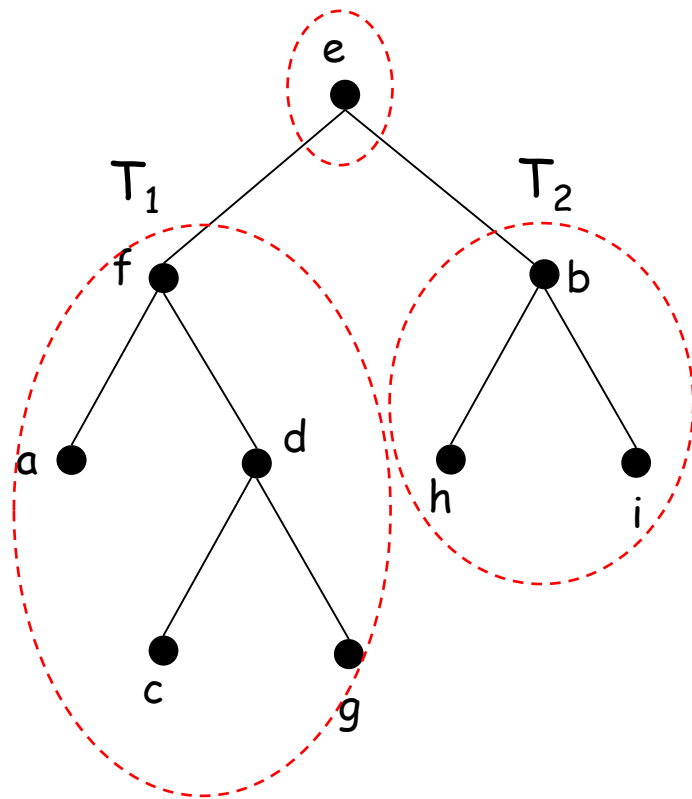
---



Recorrido en postorden:

# Árboles

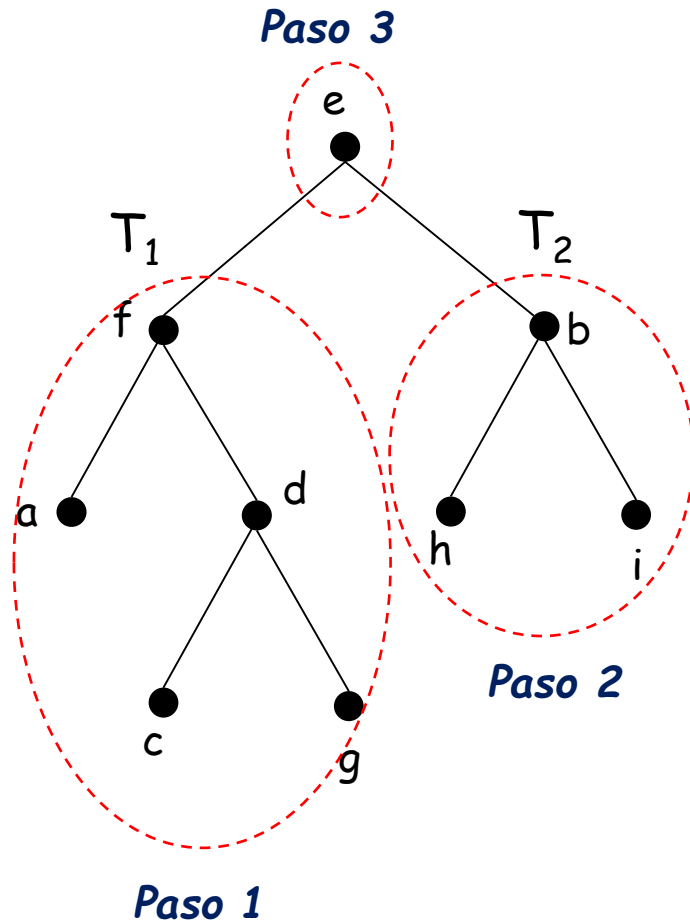
---



Recorrido en postorden:



# Árboles

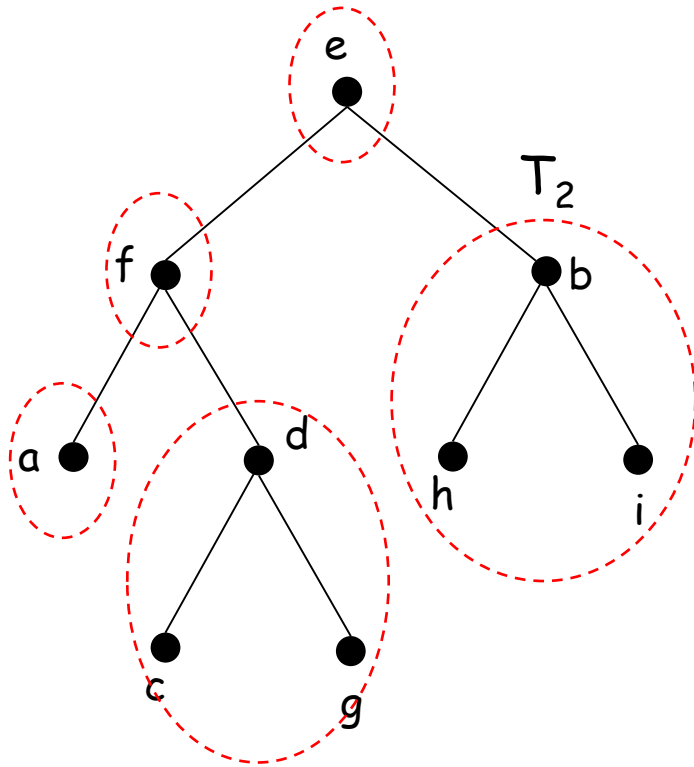


Recorrido en postorden:

# Árboles

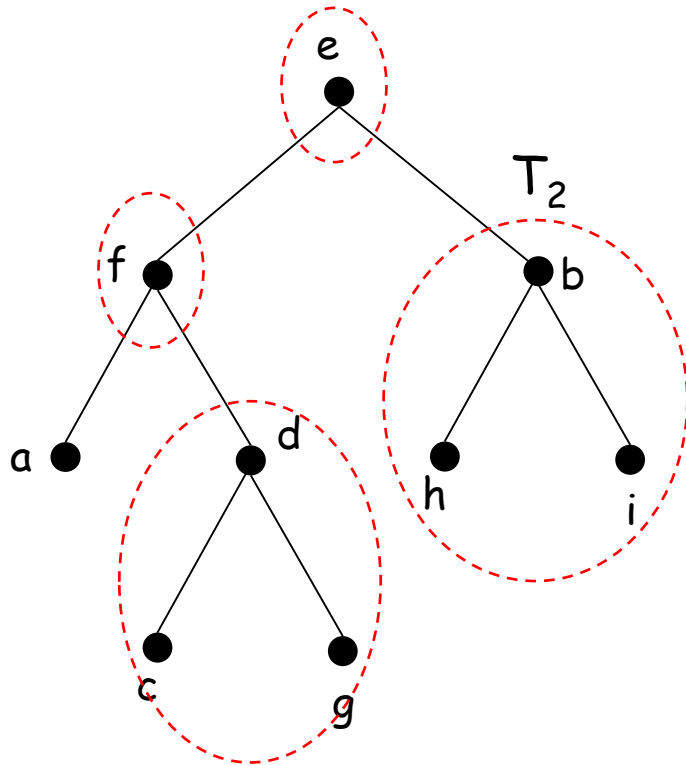
---

Recorrido en postorden:



# Árboles

---

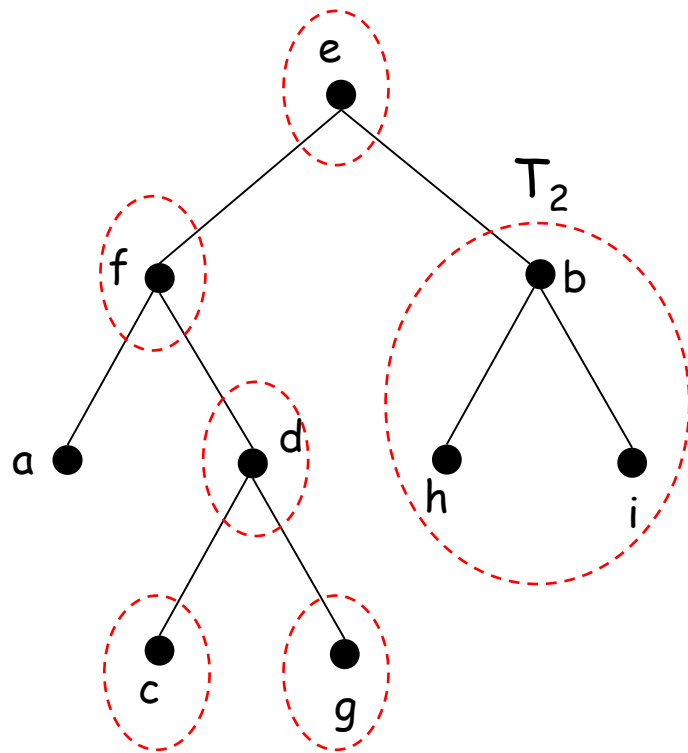


Recorrido en postorden:

a

# Árboles

---

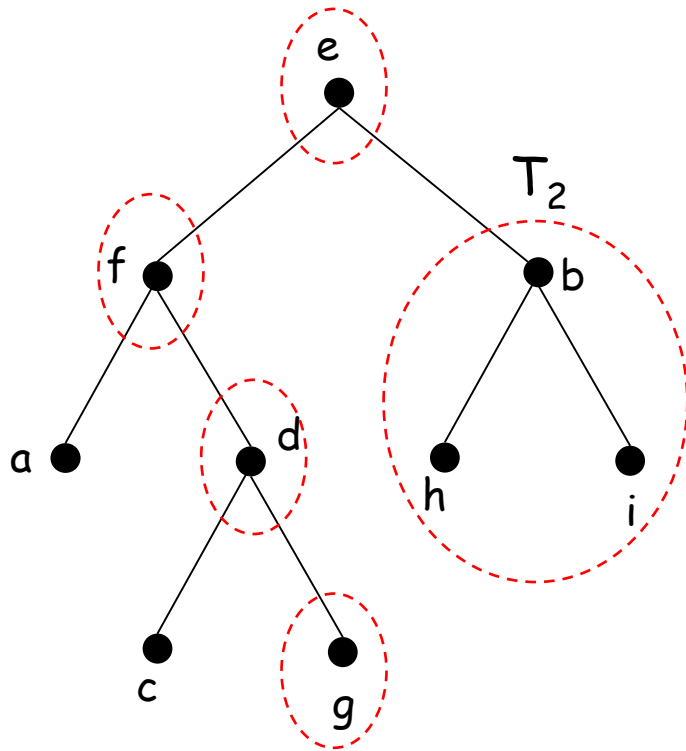


Recorrido en postorden:

a

# Árboles

---

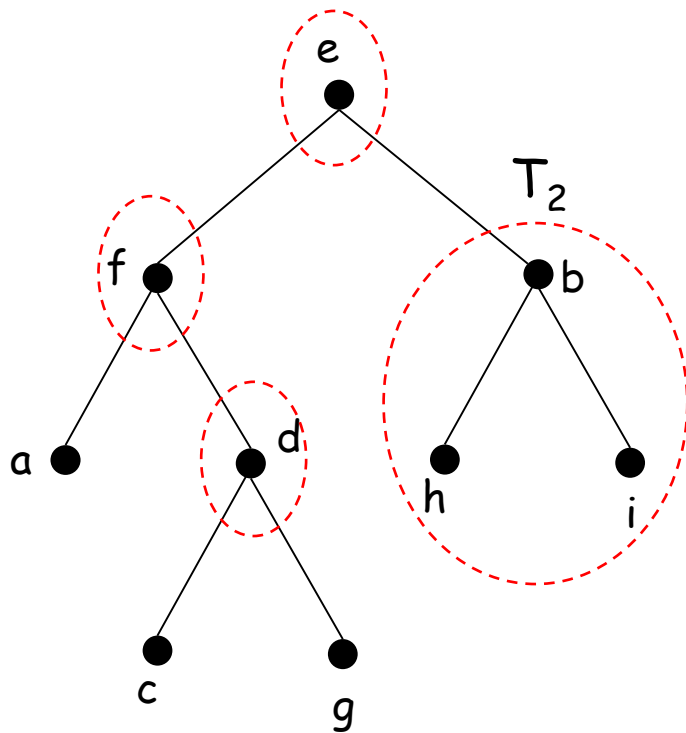


Recorrido en postorden:

a - c

# Árboles

---

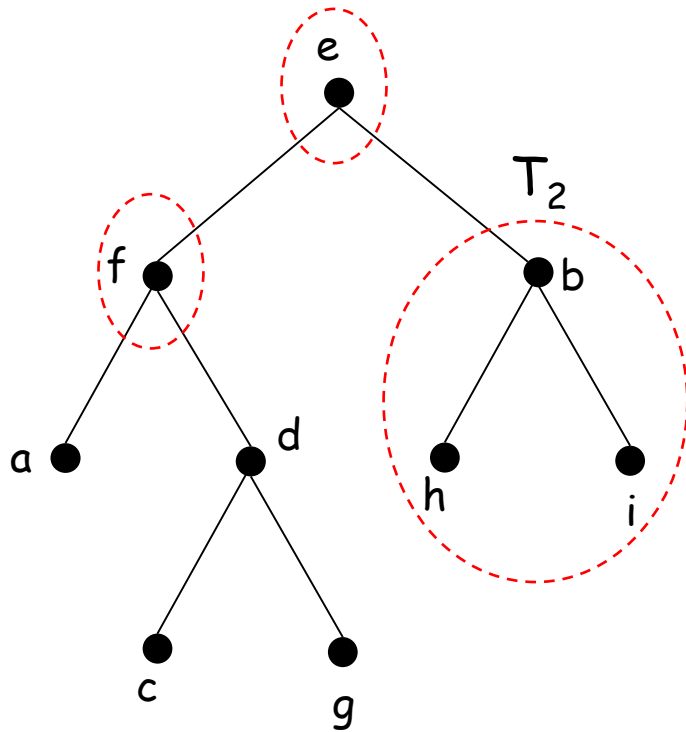


Recorrido en postorden:

a - c - g

# Árboles

---

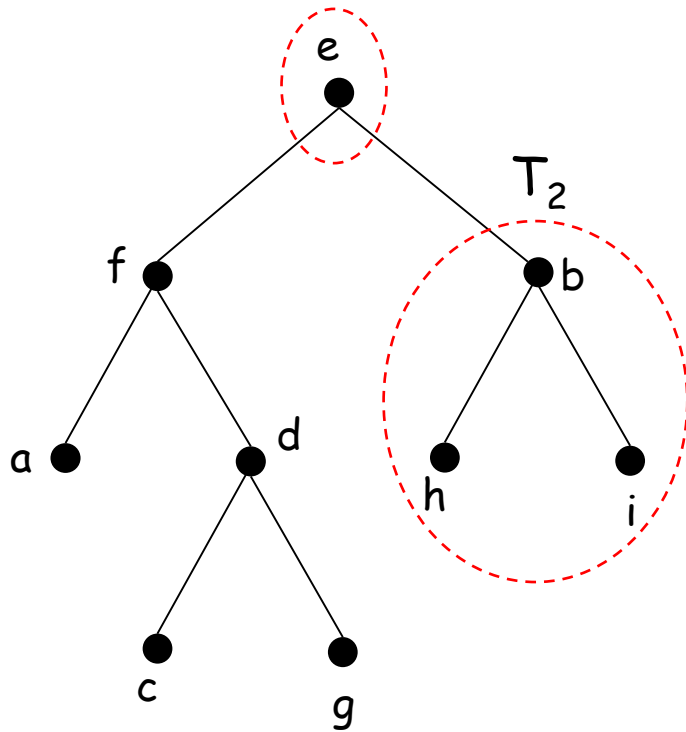


Recorrido en postorden:

a - c - g - d

# Árboles

---



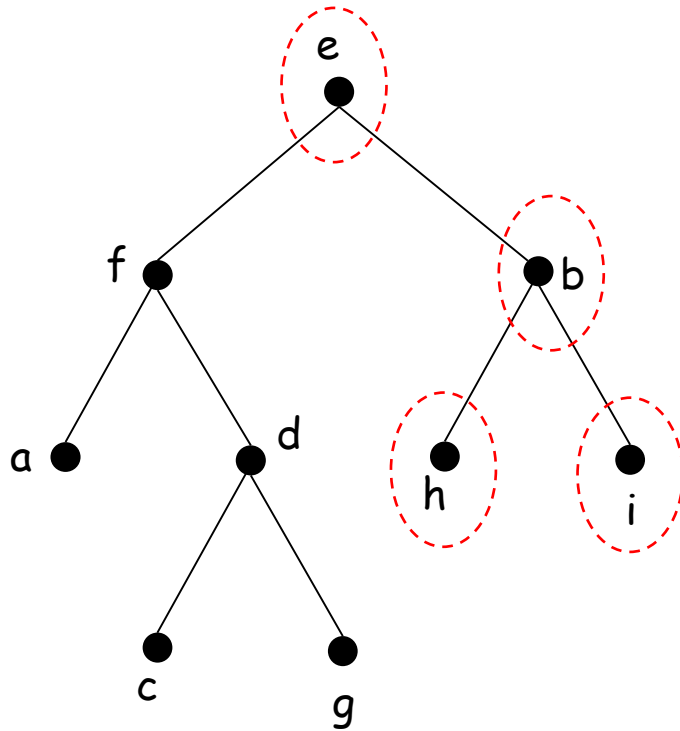
Recorrido en postorden:

a - c - g - d - f



# Árboles

---

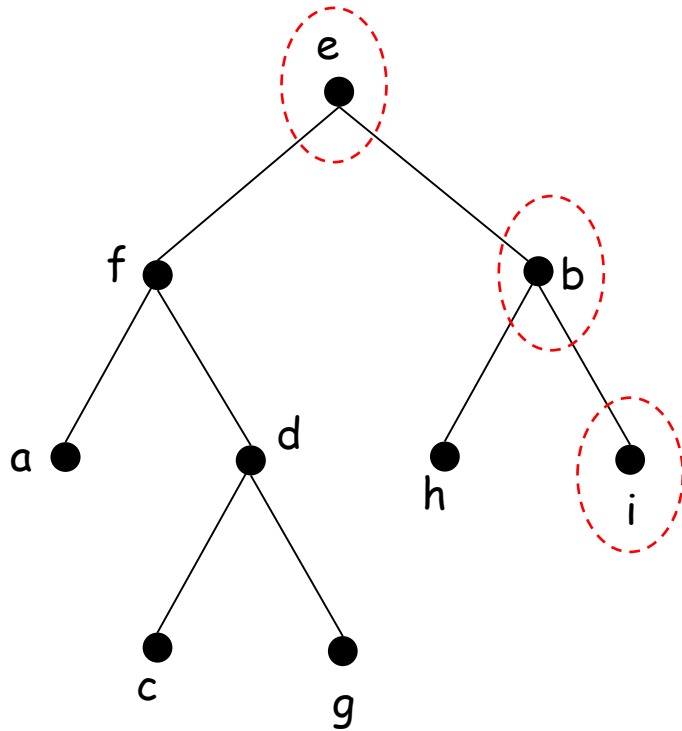


Recorrido en postorden:

a - c - g - d - f

# Árboles

---

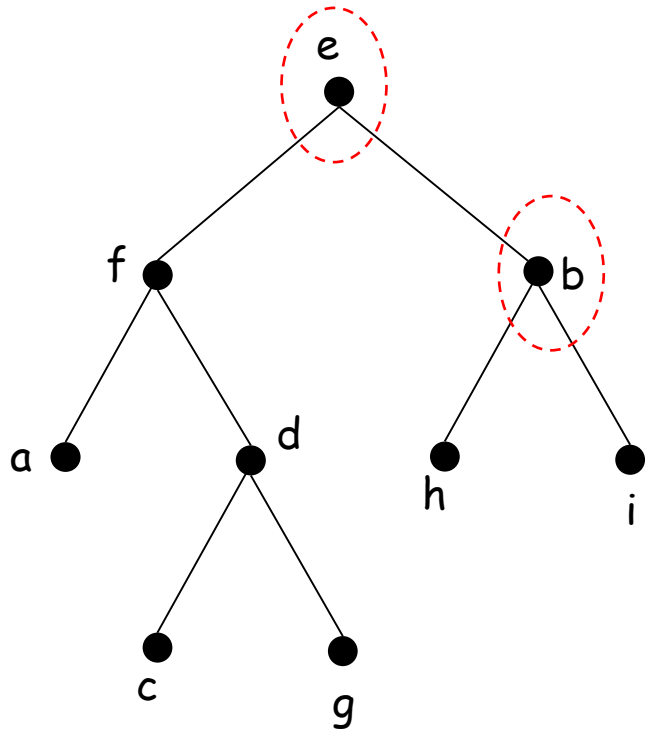


Recorrido en postorden:

a - c - g - d - f - h

# Árboles

---

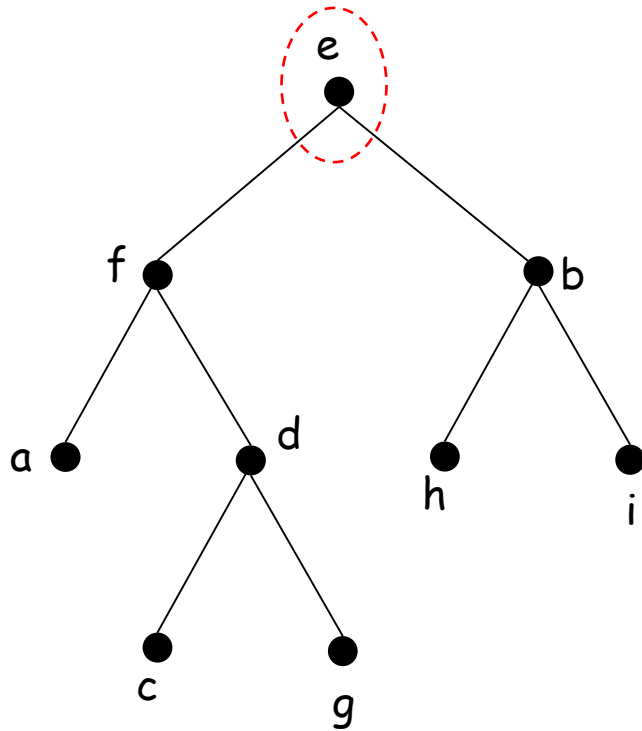


Recorrido en postorden:

a - c - g - d - f - h - i

# Árboles

---

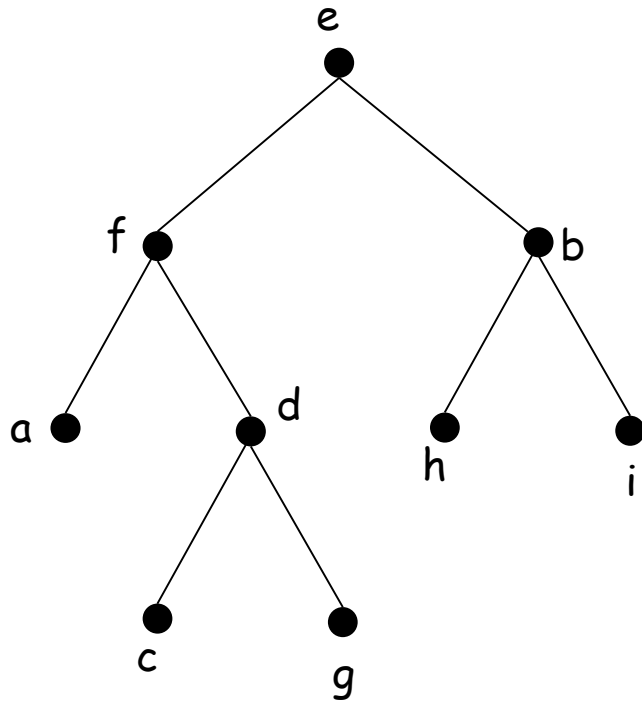


Recorrido en postorden:

a - c - g - d - f - h - i - b

# Árboles

---

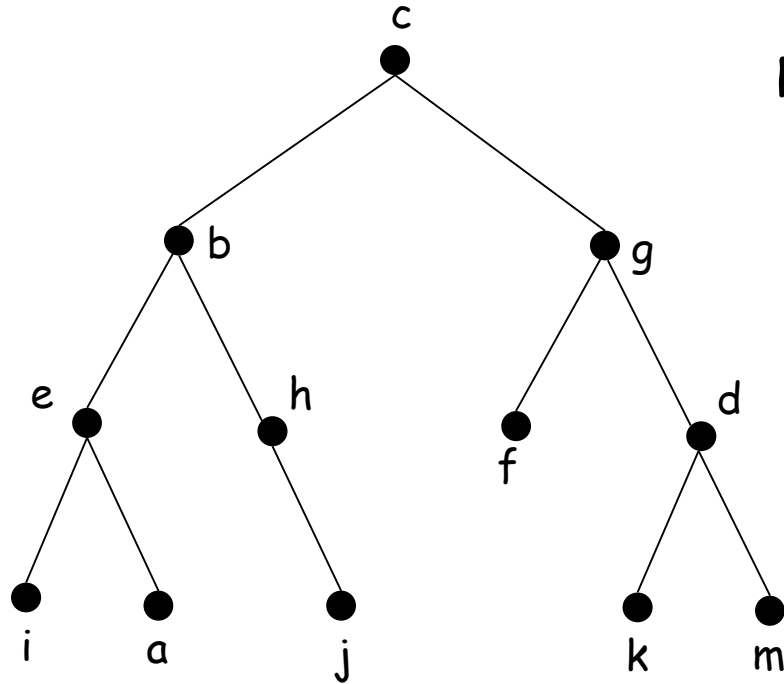


Recorrido en postorden:

a - c - g - d - f - h - i - b - e

# Árboles

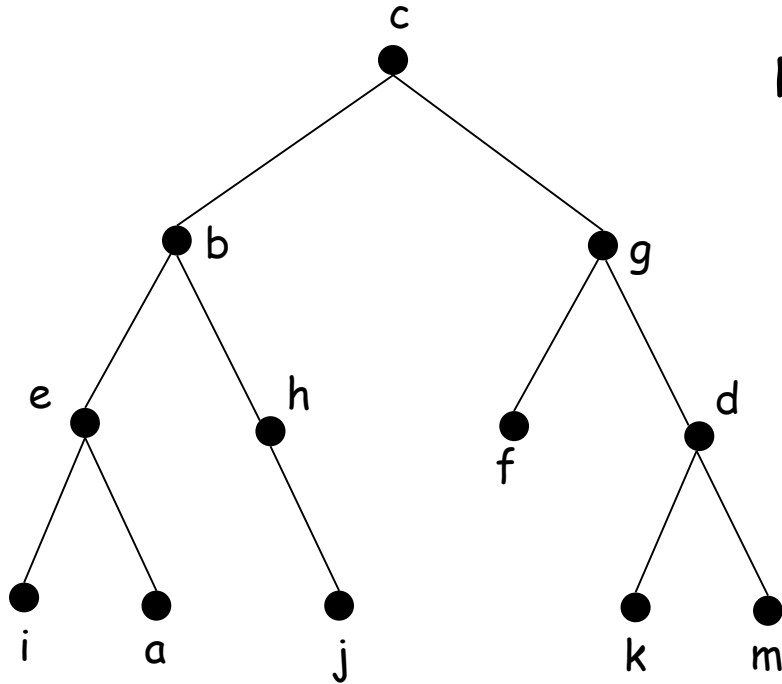
---



Recorrido en postorden:

# Árboles

---

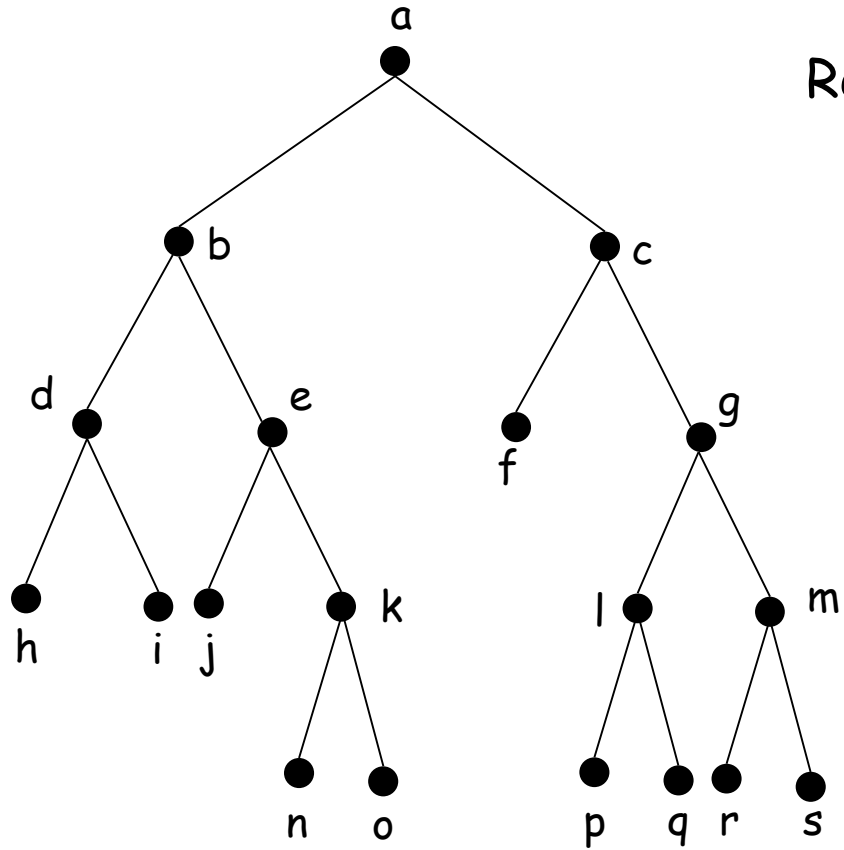


Recorrido en postorden:

i - a - e - j - h - b - f - k - m - d - g - c

# Árboles

---

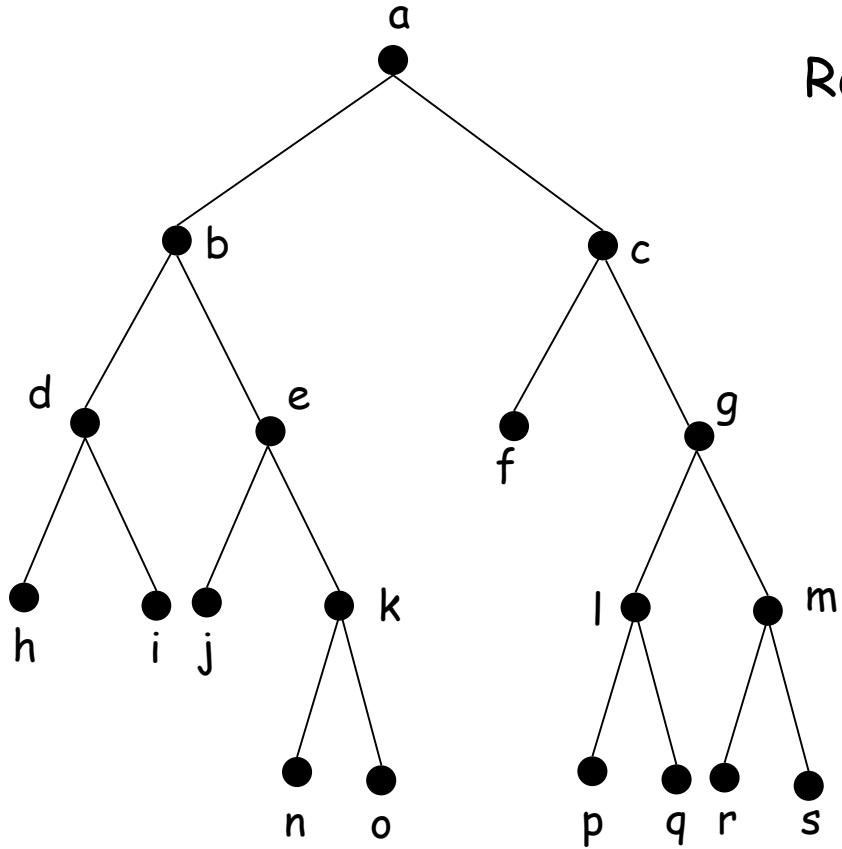


Recorrido en postorden:



# Árboles

---



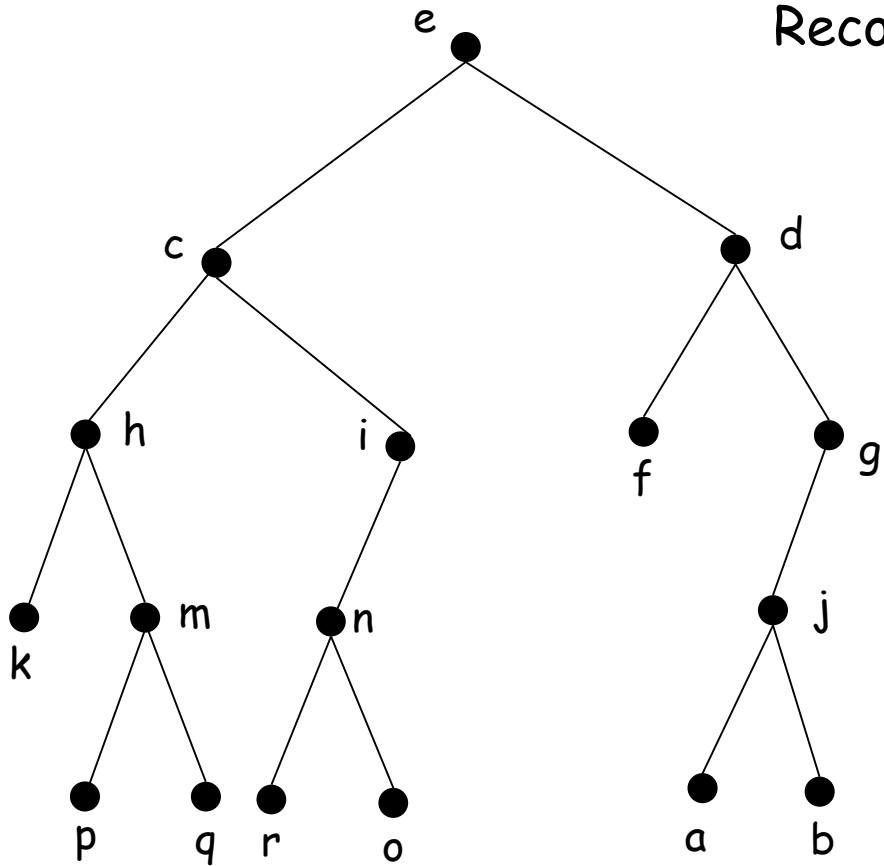
Recorrido en postorden:

h-i-d-j-n-o-k-e-b-f-p-q-l-r-s-m-g-c-a

# Árboles

---

Recorrido en postorden:

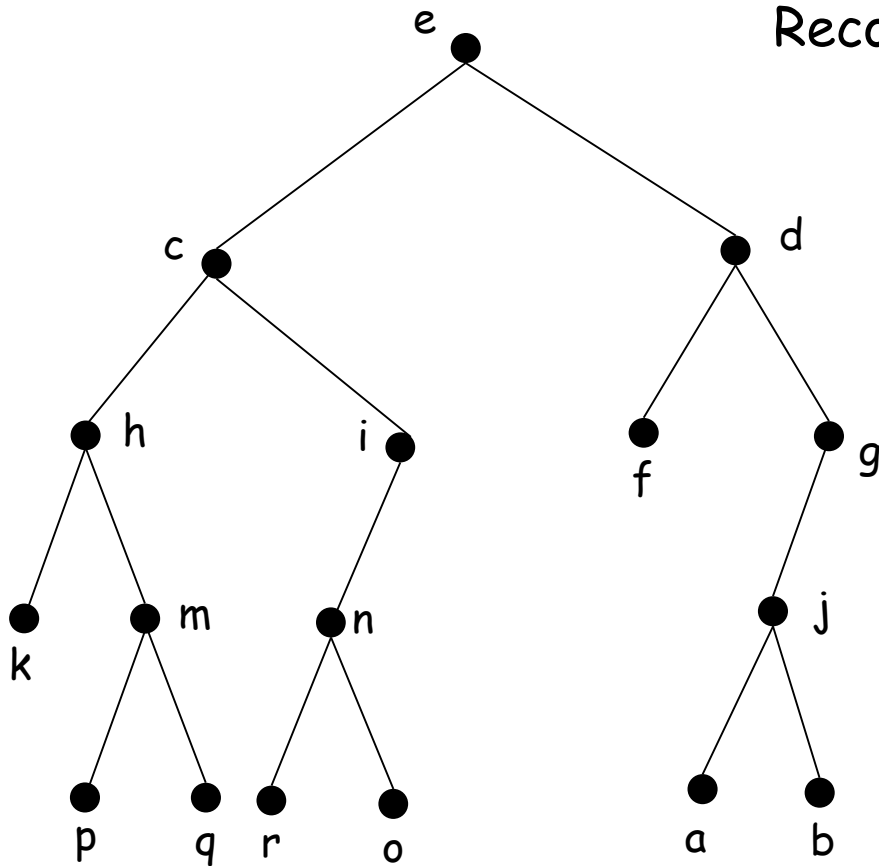


# Árboles

---

Recorrido en postorden:

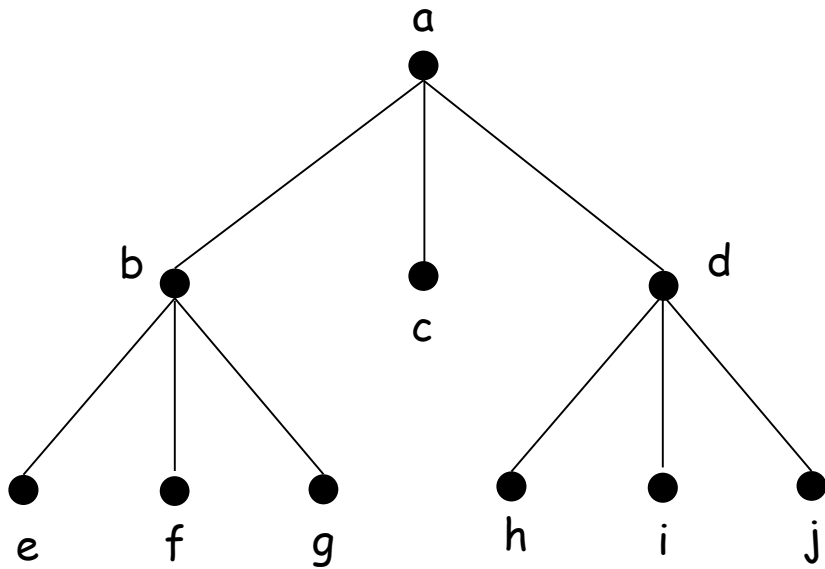
k-p-q-m-h-r-o-n-i-c-f-a-b-j-g-d-e



# Árboles

---

Recorrido en postorden:

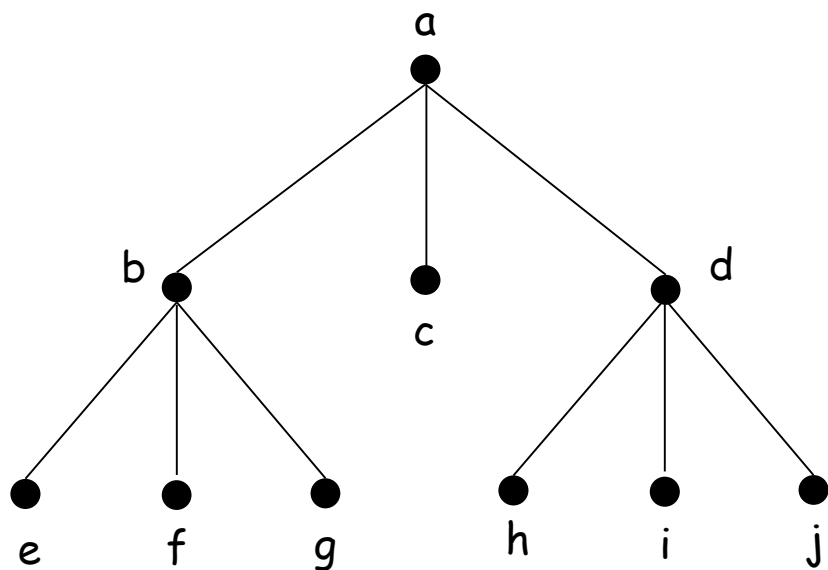


# Árboles

---

Recorrido en postorden:

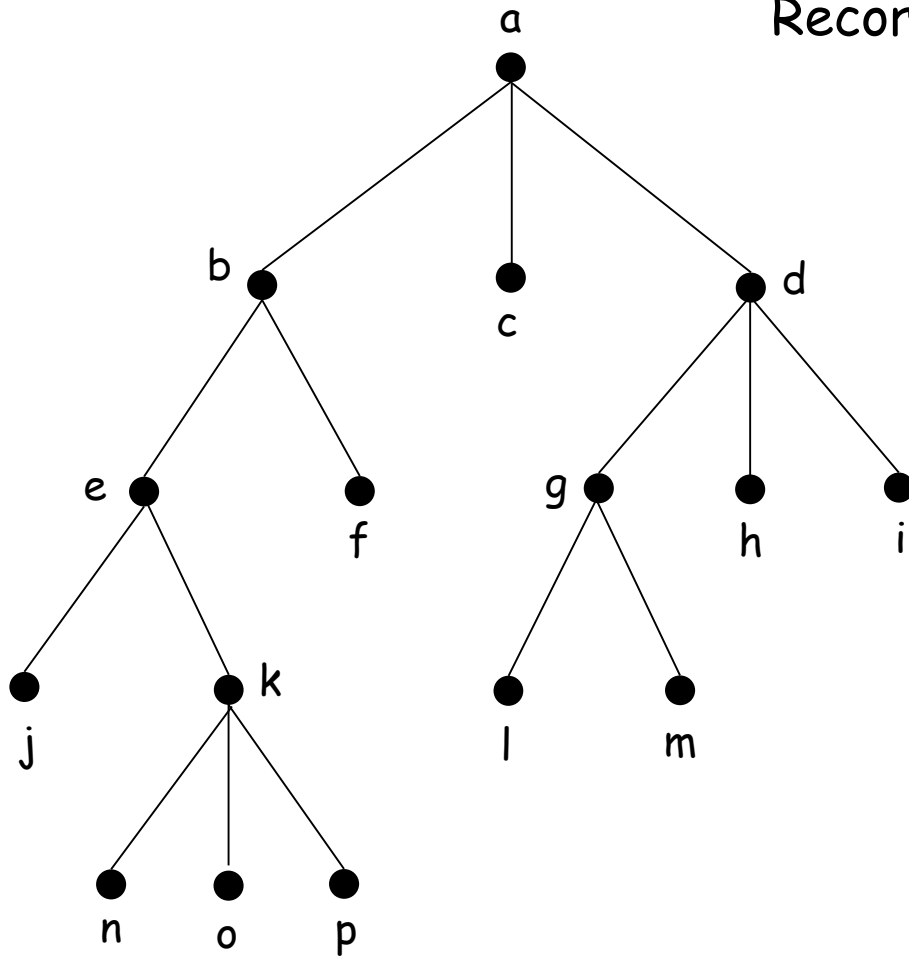
$e - f - g - b - c - h - i - j - d - a$



# Árboles

---

Recorrido en postorden:

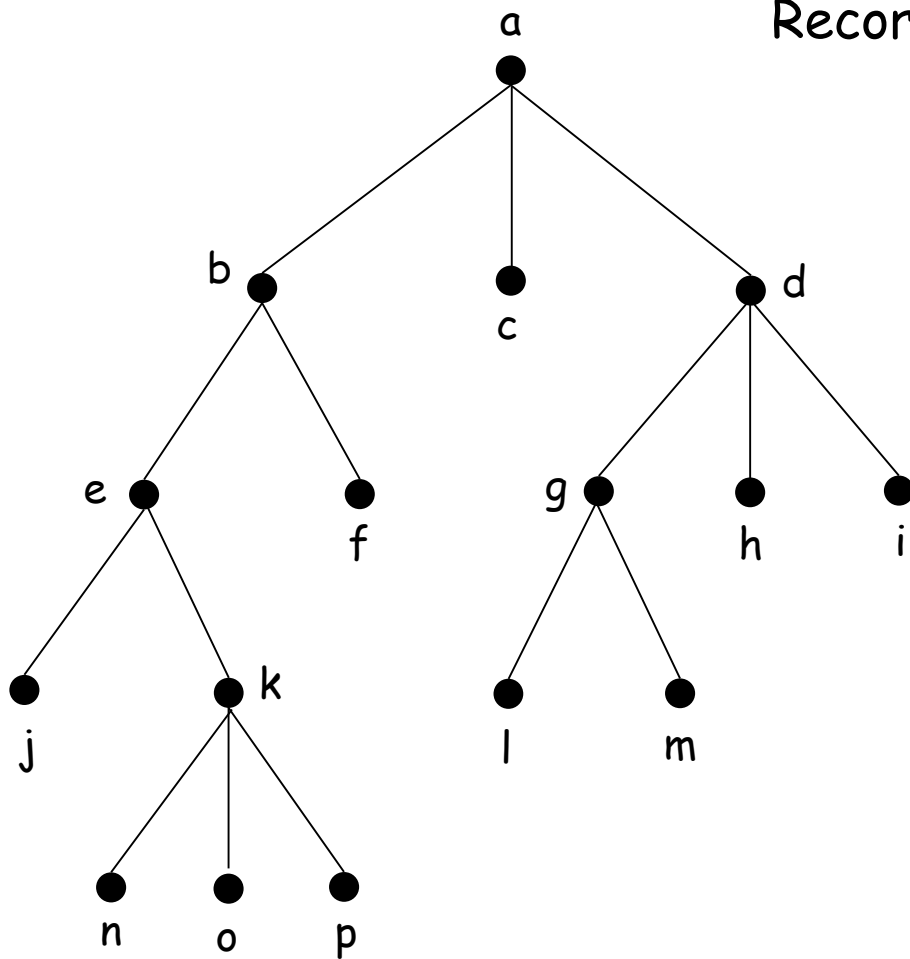


# Árboles

---

Recorrido en postorden:

j-n-o-p-k-e-f-b-c-l-m-g-h-i-d-a



# Árboles

---

## Notación infija, prefija y postfija

Permite representar expresiones complejas como proposiciones compuestas, combinaciones de conjuntos y expresiones aritméticas



# Árboles

---

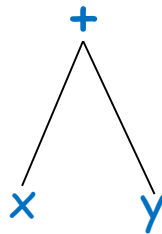
## Expresiones aritméticas

- Suma (+)
- Resta (-)
- Multiplicación (\*)
- División (/)
- Potencia ( $\uparrow$ )

# Árboles

---

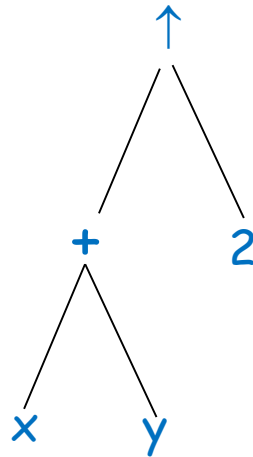
Muestre un árbol binario que represente la expresión matemática  $((x+y)^2)+((x-4)/3)$



# Árboles

---

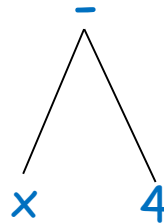
Muestre un árbol binario que represente la expresión matemática  $((x+y)^2) + ((x-4)/3)$



# Árboles

---

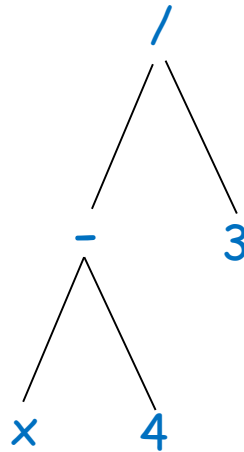
Muestre un árbol binario que represente la expresión matemática  $((x+y)^2) + ((x-4)/3)$



# Árboles

---

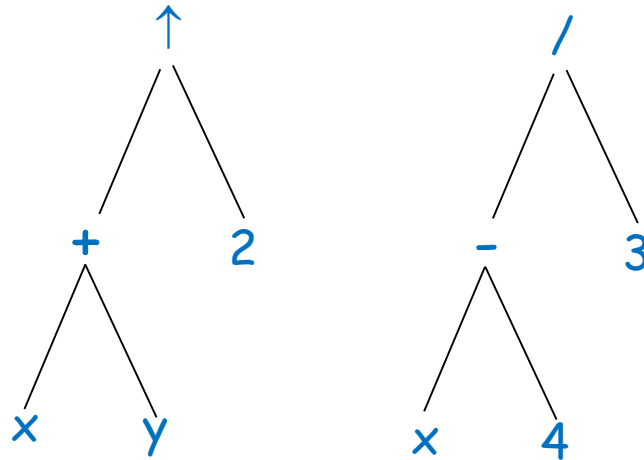
Muestre un árbol binario que represente la expresión matemática  $((x+y)^{\uparrow 2})+((x-4)/3)$



# Árboles

---

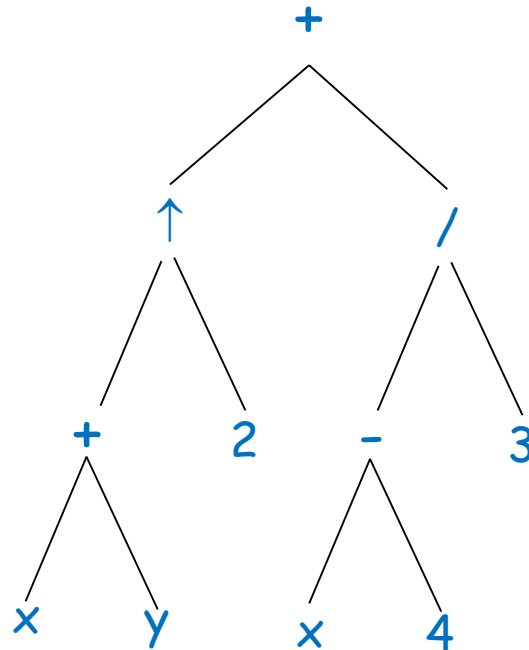
Muestre un árbol binario que represente la expresión matemática  $((x+y)^2)+((x-4)/3)$



# Árboles

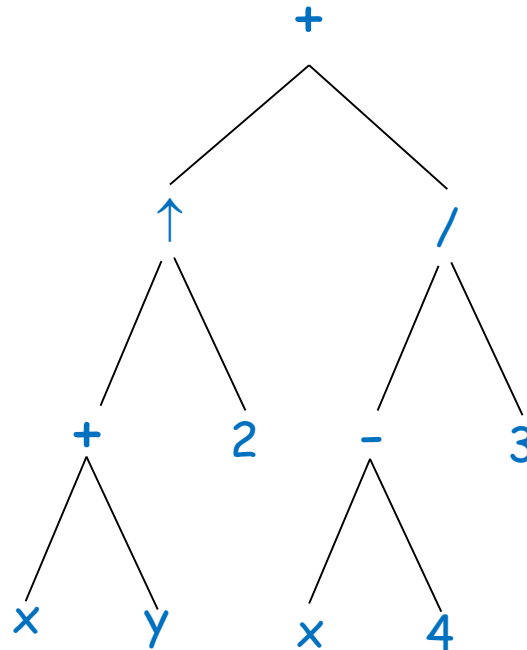
---

Muestre un árbol binario que represente la expresión matemática  $((x+y)^2)+((x-4)/3)$



# Árboles

Muestre un árbol binario que represente la expresión matemática  $((x+y)^2)+((x-4)/3)$



La expresión  $((x+y)^2)+((x-4)/3)$  se obtiene al hacer el recorrido en inorden



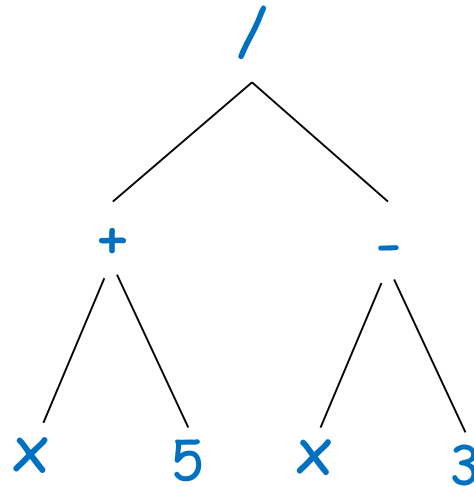
# Árboles

---

Muestre un árbol binario que represente la expresión matemática  $(x+5)/(x-3)$

# Árboles

---



Árbol binario que representa la  
expresión matemática  $(x+5)/(x-3)$

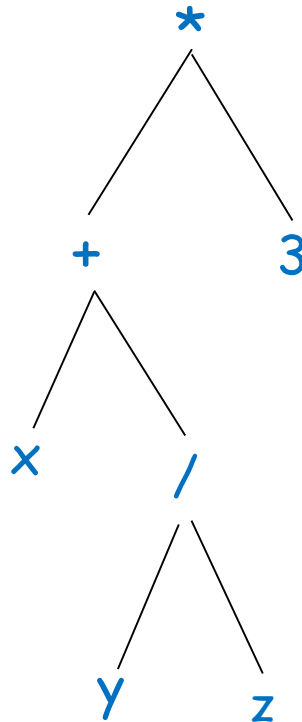
# Árboles

---

Muestre un árbol binario que represente la expresión matemática  $(x+(y/z))^*3$

# Árboles

---



Árbol binario que representa la  
expresión matemática  $(x + (y / z)) * 3$

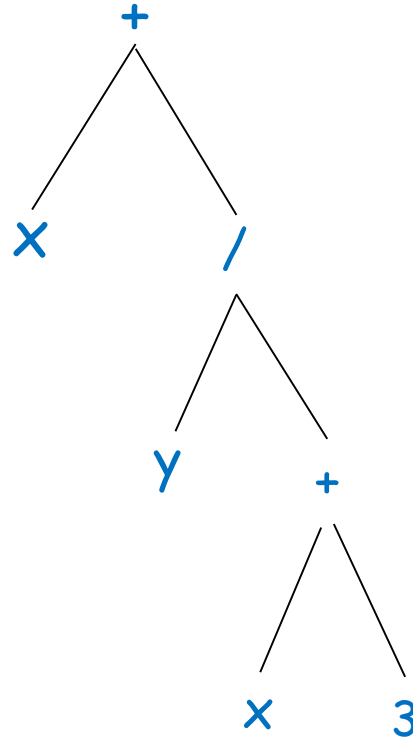
# Árboles

---

Muestre un árbol binario que represente la expresión matemática  $x + (y / (x + 3))$

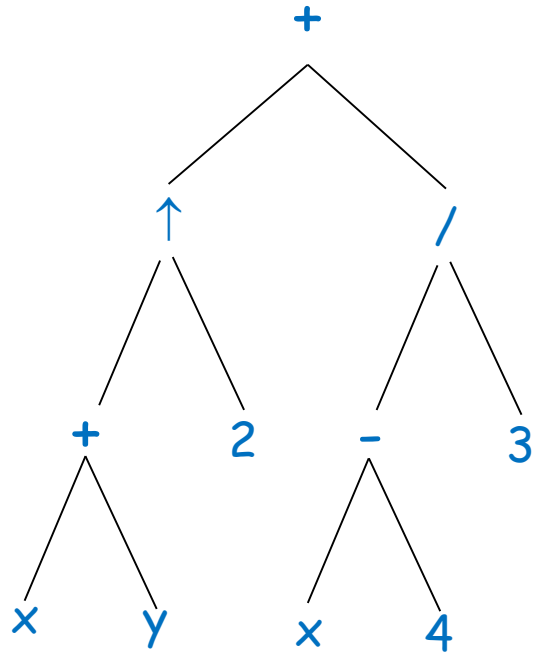
# Árboles

---

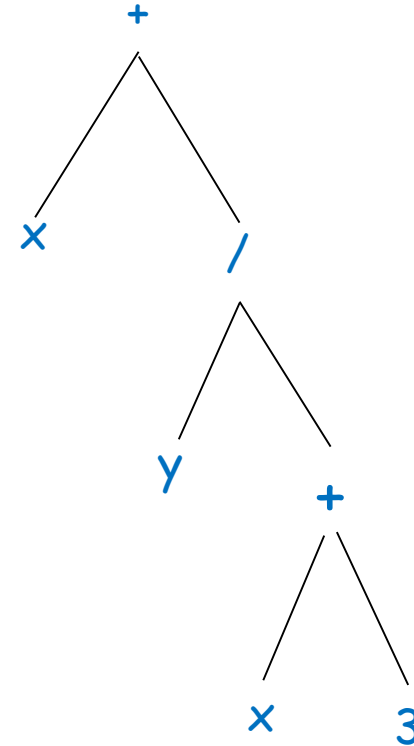


Árbol binario que representa la  
expresión matemática  $x + (y / (x + 3))$

# Árboles



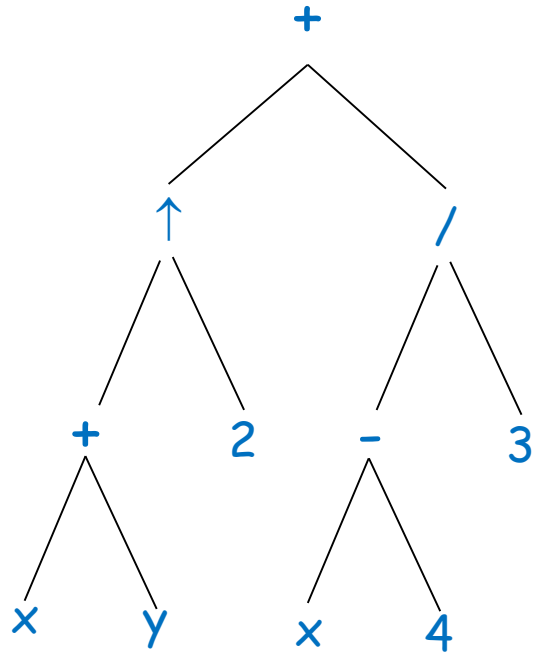
Árbol binario que representa  
la expresión matemática  
 $((x+y)^{\uparrow 2}) + ((x-4)/3)$



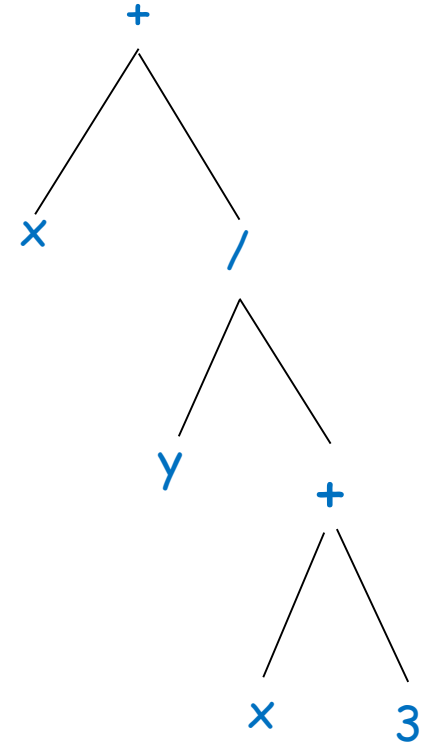
Árbol binario que representa  
la expresión matemática  
 $x + (y / (x + 3))$

# Árboles

---



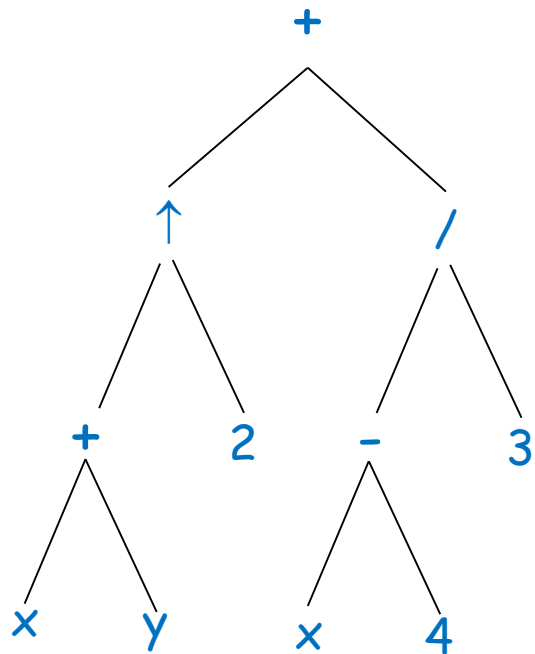
Recorrido en inorden:



Recorrido en inorden:

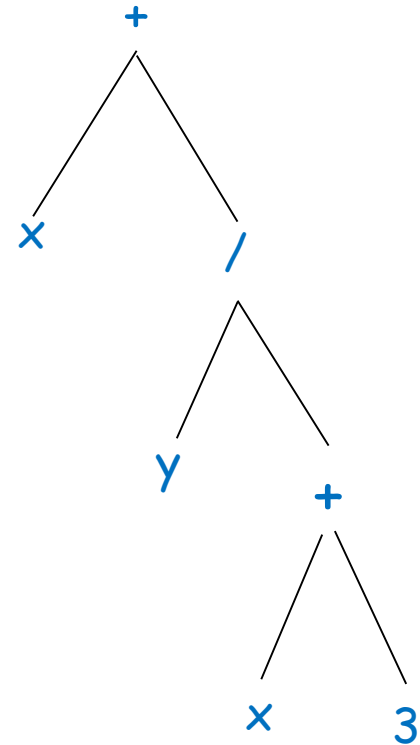


# Árboles



Recorrido en inorden:

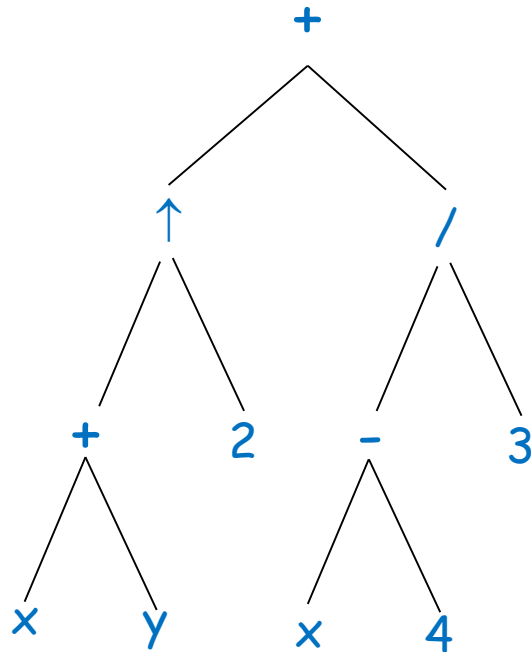
$$((x+y)^2)+((x-4)/3)$$



Recorrido en inorden:

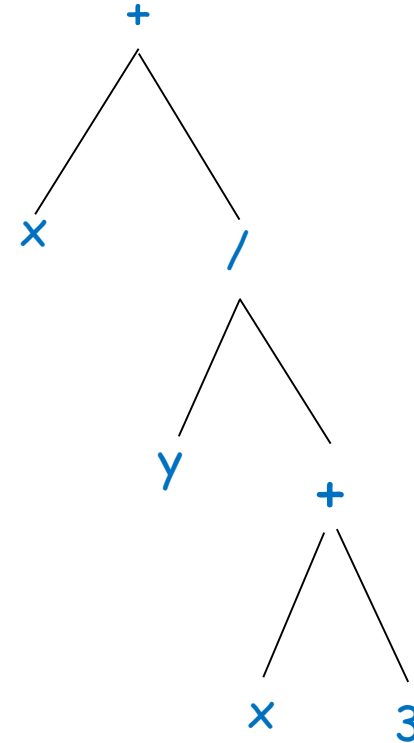
$$x+(y/(x+3))$$

# Árboles



Recorrido en inorden:

$((x+y)^2)+((x-4)/3)$



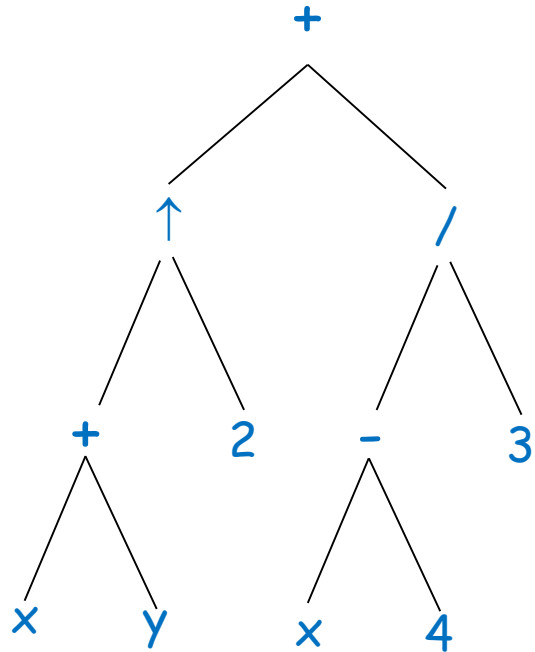
Recorrido en inorden:

$x+(y/(x+3))$

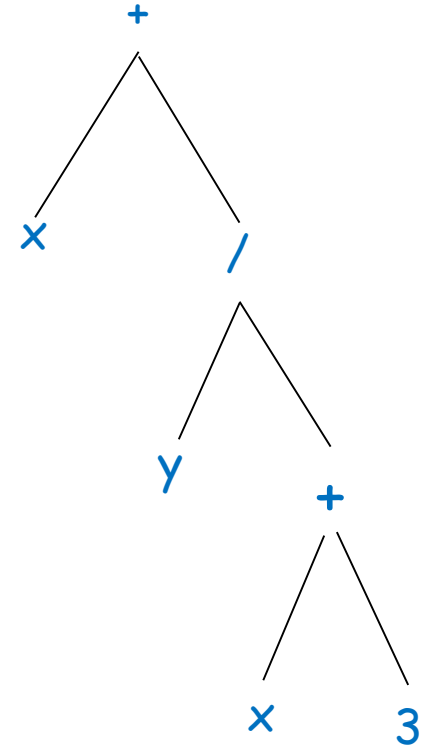
**Notación infija**

# Árboles

---

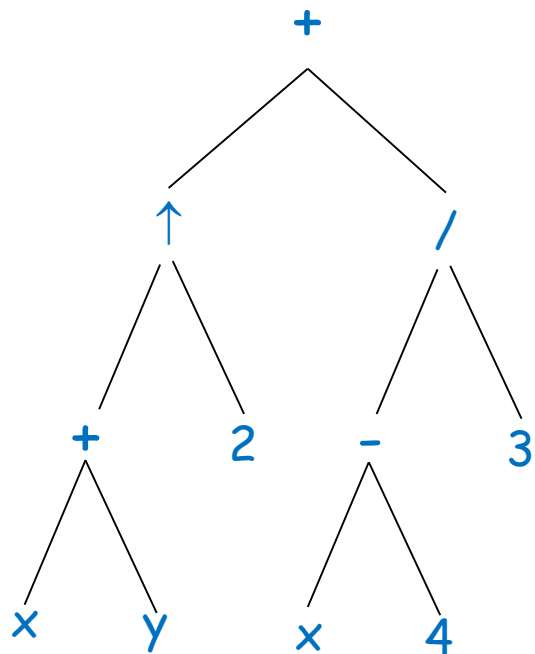


Recorrido en preorden:



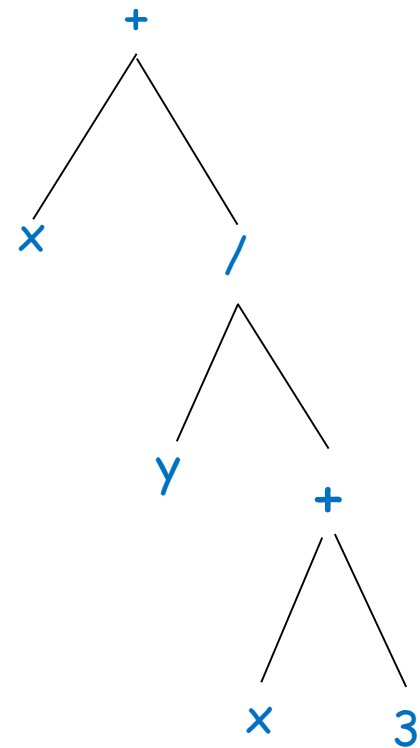
Recorrido en preorden:

# Árboles



Recorrido en preorden:

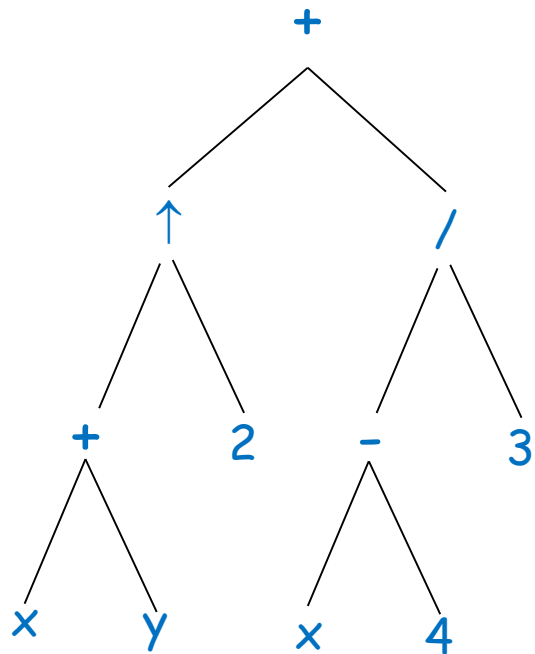
+ ↑ + x y 2 / - x 4 3



Recorrido en preorden:

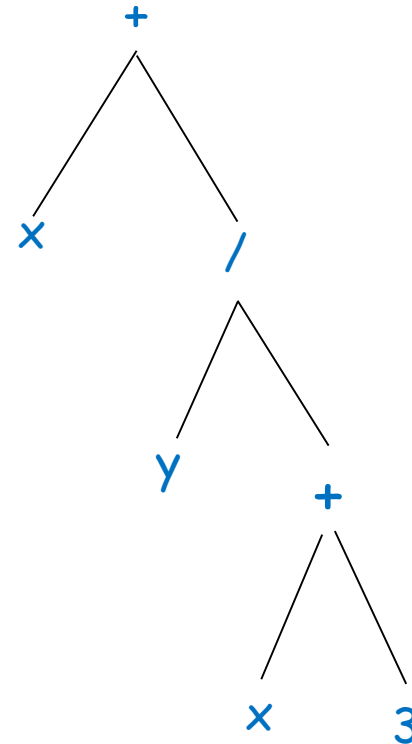
+ x / y + x 3

# Árboles



Recorrido en preorden:

+ ↑ + x y 2 / - x 4 3



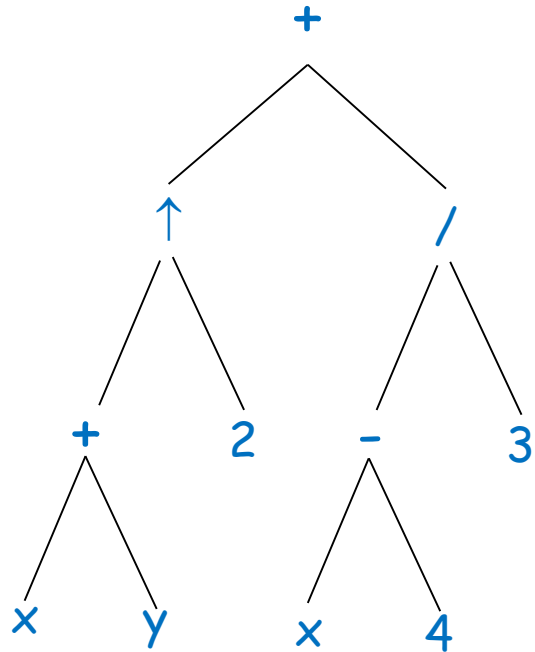
Recorrido en preorden:

+ x / y + x 3

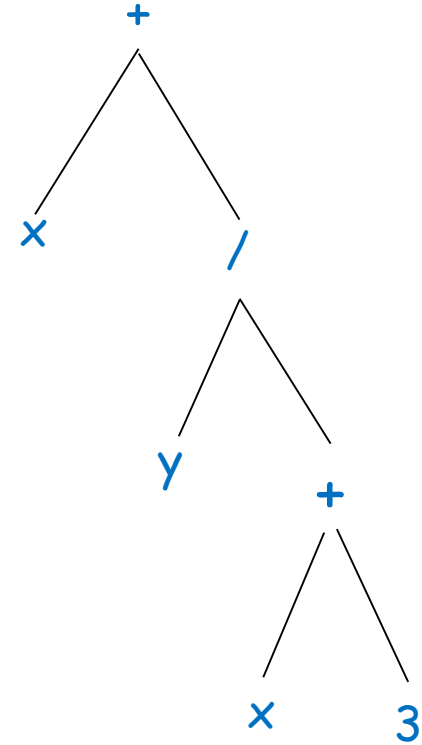
Notación prefija

# Árboles

---

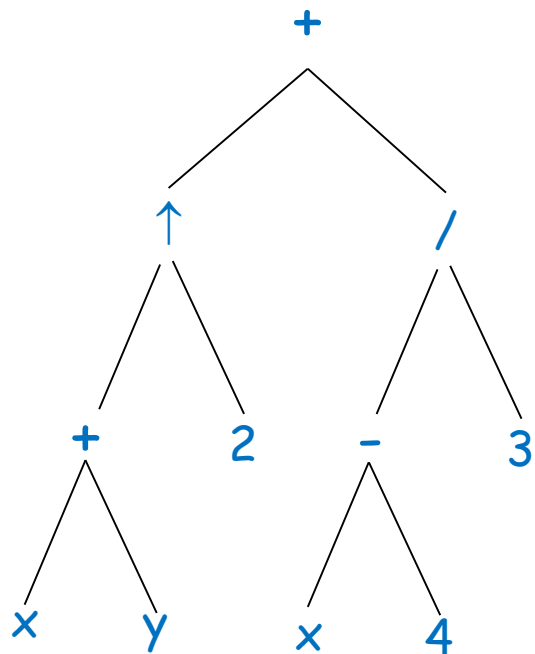


Recorrido en postorden:



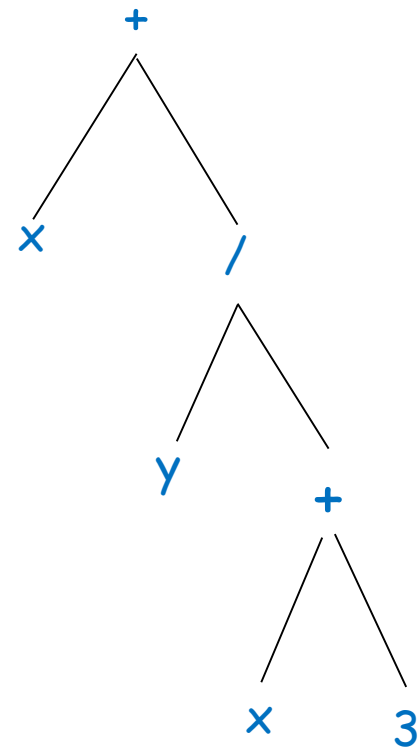
Recorrido en postorden:

# Árboles



Recorrido en postorden:

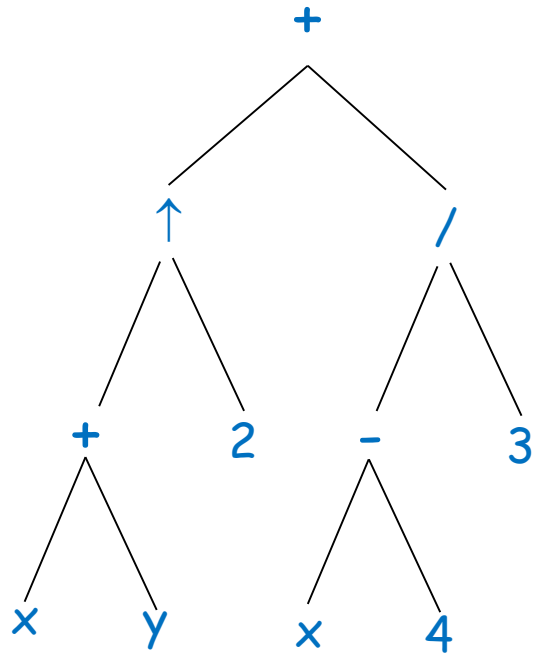
$x\ y\ +\ 2\ \uparrow\ x\ 4\ -\ 3\ /\ +$



Recorrido en postorden:

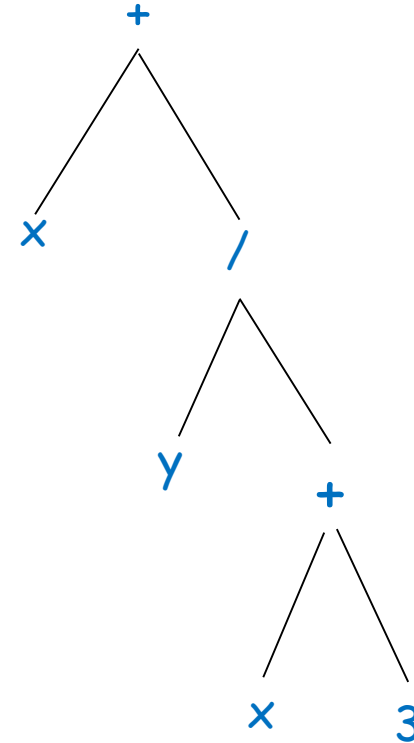
$x\ y\ x\ 3\ +\ /\ +$

# Árboles



Recorrido en postorden:

$x y + 2 \uparrow x 4 - 3 / +$



Recorrido en postorden:

$x y x 3 + / +$

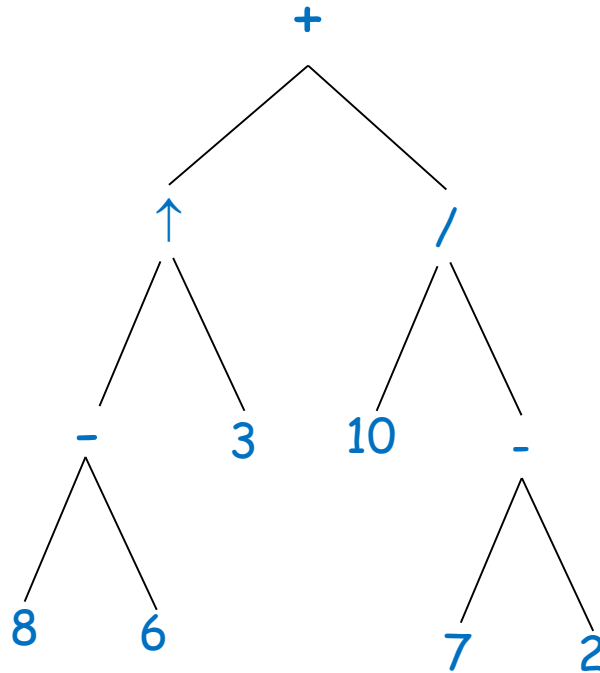
**Notación postfija**



# Árboles

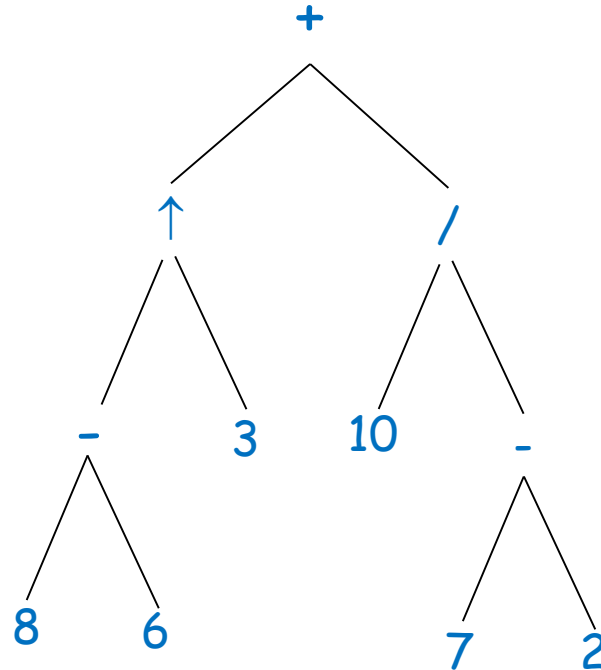
---

Representar en notación infija, prefija y postfija, la expresión matemática dada por el siguiente árbol:



# Árboles

---



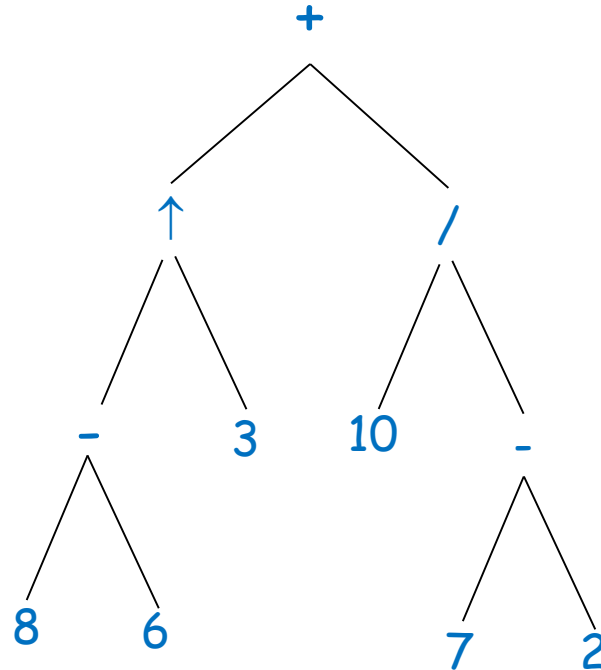
Notación **infija**:  $((8-6)\uparrow 3)+(10/(7-2))$

Notación **prefija**:  $+ \uparrow - 8 6 3 / 10 - 7 2$

Notación **postfija**:  $8 6 - 3 \uparrow 10 7 2 - / +$

# Árboles

---



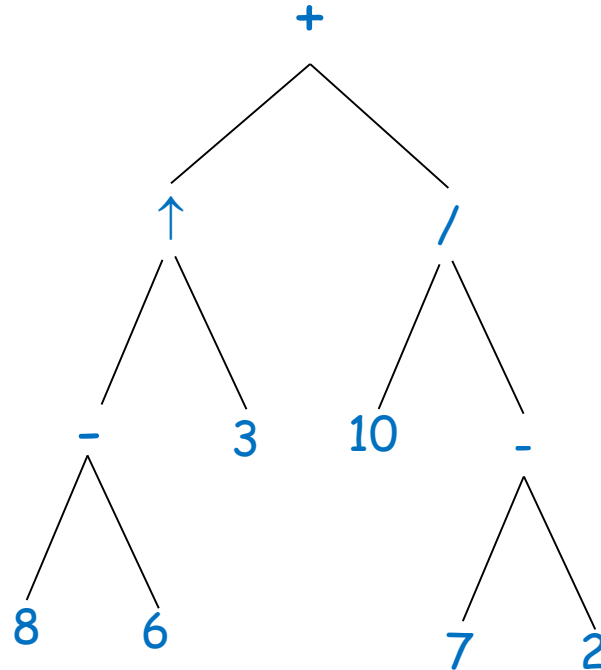
Notación **infija**:  $((8-6)\uparrow 3)+(10/(7-2)) = ?$

Notación **prefija**:  $+ \uparrow - 8 6 3 / 10 - 7 2$

Notación **postfija**:  $8 6 - 3 \uparrow 10 7 2 - / +$

# Árboles

---



Notación **infija**:  $((8-6)\uparrow 3)+(10/(7-2)) = 10$

Notación **prefija**:  $+ \uparrow - 8 6 3 / 10 - 7 2$

Notación **postfija**:  $8 6 - 3 \uparrow 10 7 2 - / +$

# Árboles

---

Notación **prefija**: + ↑ - 8 6 3 / 10 - 7 2

# Árboles

---

Notación **prefija**: + ↑ - 8 6 3 / 10 - 7 2

# Árboles

---

Notación **prefija**: + ↑ - 8 6 3 / 10 - 7 2

+ ↑ - 8 6 3 / 10 5

# Árboles

---

Notación **prefija**: + ↑ - 8 6 3 / 10 - 7 2

+ ↑ - 8 6 3 / 10 5



# Árboles

---

Notación **prefija**: + ↑ - 8 6 3 / 10 - 7 2

+ ↑ - 8 6 3 / 10 5

+ ↑ - 8 6 3 2

# Árboles

---

Notación **prefija**: + ↑ - 8 6 3 / 10 - 7 2

+ ↑ - 8 6 3 / 10 5

+ ↑ - 8 6 3 2

# Árboles

---

Notación **prefija**: + ↑ - 8 6 3 / 10 - 7 2

+ ↑ - 8 6 3 / 10 5

+ ↑ - 8 6 3 2

+ ↑ 2 3 2

# Árboles

---

Notación **prefija**: + ↑ - 8 6 3 / 10 - 7 2

+ ↑ - 8 6 3 / 10 5

+ ↑ - 8 6 3 2

+ ↑ 2 3 2

# Árboles

---

Notación **prefija**: + ↑ - 8 6 3 / 10 - 7 2

+ ↑ - 8 6 3 / 10 5

+ ↑ - 8 6 3 2

+ ↑ 2 3 2

+ 8 2

# Árboles

---

Notación **prefija**: + ↑ - 8 6 3 / 10 - 7 2

+ ↑ - 8 6 3 / 10 5

+ ↑ - 8 6 3 2

+ ↑ 2 3 2

+ 8 2

# Árboles

---

Notación **prefija**: + ↑ - 8 6 3 / 10 - 7 2

+ ↑ - 8 6 3 / 10 5

+ ↑ - 8 6 3 2

+ ↑ 2 3 2

+ 8 2

10

# Árboles

---

Notación **postfija**: 8 6 - 3 ↑ 10 7 2 - / +



# Árboles

---

Notación **postfija**: 8 6 - 3 ↑ 10 7 2 - / +

# Árboles

---

Notación **postfija**: 8 6 - 3 ↑ 10 7 2 - / +

2 3 ↑ 10 7 2 - / +

# Árboles

---

Notación **postfija**: 8 6 - 3 ↑ 10 7 2 - / +

2 3 ↑ 10 7 2 - / +

# Árboles

---

Notación **postfija**:  $\underbrace{8\ 6\ -}_3\ \uparrow\ 10\ 7\ 2\ -\ /\ +$

$\underbrace{2\ 3\ \uparrow}_{10}\ 7\ 2\ -\ /\ +$

$8\ 10\ 7\ 2\ -\ /\ +$

# Árboles

---

Notación **postfija**:  $\underbrace{8\ 6\ -}_3\ \uparrow\ 10\ 7\ 2\ -\ /\ +$

$\underbrace{2\ 3\ \uparrow}_{10}\ 7\ 2\ -\ /\ +$

$8\ 10\ \underbrace{7\ 2\ -}_{10}\ /\ +$

# Árboles

---

Notación **postfija**:  $\underline{8\ 6\ -}\ 3\ \uparrow\ 10\ 7\ 2\ -\ /\ +$

$\underline{2\ 3\ \uparrow}\ 10\ 7\ 2\ -\ /\ +$

$8\ 10\ \underline{7\ 2\ -}\ /\ +$

$8\ 10\ 5\ /\ +$

# Árboles

---

Notación **postfija**:  $\underbrace{8\ 6\ -}_3\ \uparrow\ 10\ 7\ 2\ -\ /\ +$

$\underbrace{2\ 3\ \uparrow}_{10}\ 7\ 2\ -\ /\ +$

$8\ 10\ \underbrace{7\ 2\ -}_{10}\ /\ +$

$8\ \underbrace{10\ 5\ /\ }_{10}\ +$

# Árboles

---

Notación **postfija**:  $\underbrace{8\ 6\ -}_3\ \uparrow\ 10\ 7\ 2\ -\ /\ +$

$\underbrace{2\ 3\ \uparrow}_{10}\ 7\ 2\ -\ /\ +$

$8\ 10\ \underbrace{7\ 2\ -}_{10}\ /\ +$

$8\ \underbrace{10\ 5\ /\ }_{10}\ +$

$8\ 2\ +$



# Árboles

---

Notación **postfija**:  $\underbrace{8\ 6\ -}_3\ \uparrow\ 10\ 7\ 2\ -\ /\ +$

$\underbrace{2\ 3\ \uparrow}_{10}\ 7\ 2\ -\ /\ +$

$8\ 10\ \underbrace{7\ 2\ -}_{10}\ /\ +$

$8\ \underbrace{10\ 5\ /\ }_{10}\ +$

$\underbrace{8\ 2\ +}_{10}$

# Árboles

---

Notación **postfija**:  $\underbrace{8\ 6\ -}_3\ \uparrow\ 10\ 7\ 2\ -\ /\ +$

$\underbrace{2\ 3\ \uparrow}_8\ 10\ 7\ 2\ -\ /\ +$

$8\ 10\ \underbrace{7\ 2\ -}_{10}\ /\ +$

$8\ \underbrace{10\ 5\ /\ }_{10}\ +$

$\underbrace{8\ 2\ +}_{10}$

10

# Árboles

---

Indique el valor de la siguiente expresión que está en notación prefija

Notación **prefija**: + - \* 2 3 5 / ↑ 2 3 4

# Árboles

---

Indique el valor de la siguiente expresión que está en notación prefija

Notación **prefija**: + - \* 2 3 5 / ↑ 2 3 4

+ - \* 2 3 5 / 8 4

+ - \* 2 3 5 2

+ - 6 5 2

+ 1 2

3

# Árboles

---

Indique el valor de la siguiente expresión que está en notación postfija

Notación **postfija**:  $7\ 2\ 3\ *\ -\ 4\ \uparrow\ 9\ 3\ /\ +$

# Árboles

---

Indique el valor de la siguiente expresión que está en notación postfija

Notación **postfija**:  $7 \ 2 \ 3 \ * \ - \ 4 \ \uparrow \ 9 \ 3 \ / \ +$

$7 \ 6 \ - \ 4 \ \uparrow \ 9 \ 3 \ / \ +$

$1 \ 4 \ \uparrow \ 9 \ 3 \ / \ +$

$1 \ 9 \ 3 \ / \ +$

$1 \ 3 \ +$

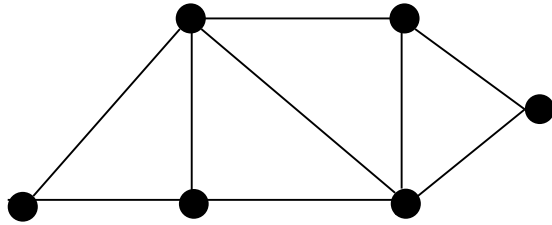
4

# Árboles

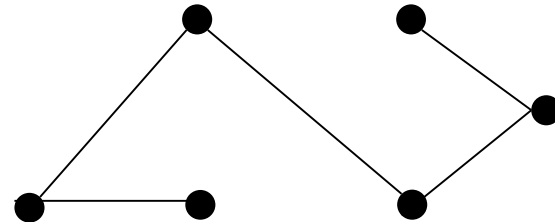
---

## Árbol recubridor

Sea  $G$  un grafo simple, un árbol recubridor de  $G$  es un subgrafo de  $G$  que es un árbol y contiene todos los vértices de  $G$



Grafo  $G$

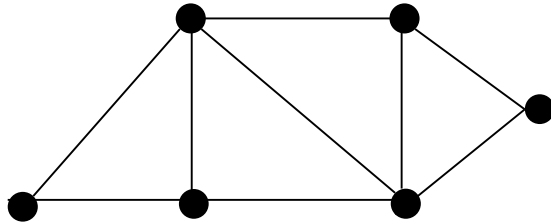


Árbol recubridor de  $G$

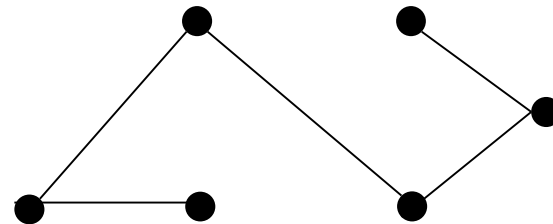
# Árboles

## Árbol recubridor

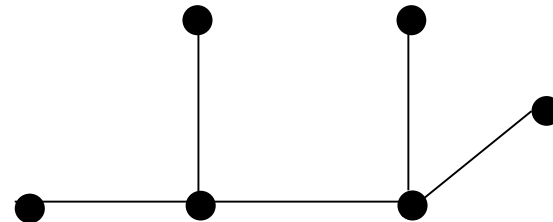
Sea  $G$  un grafo simple, un árbol recubridor de  $G$  es un subgrafo de  $G$  que es un árbol y contiene todos los vértices de  $G$



Grafo  $G$



Árbol recubridor de  $G$

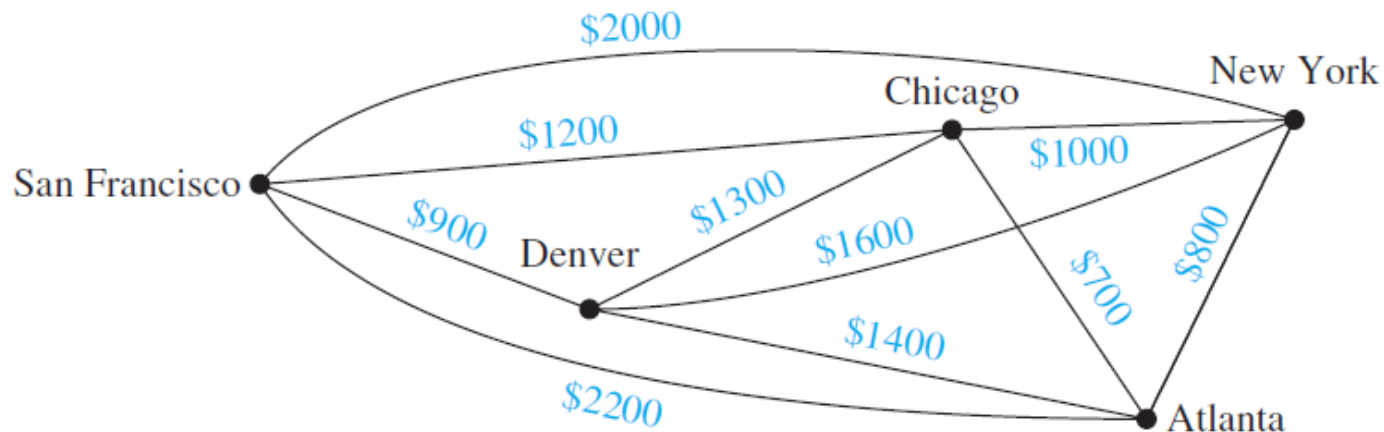


Árbol recubridor de  $G$



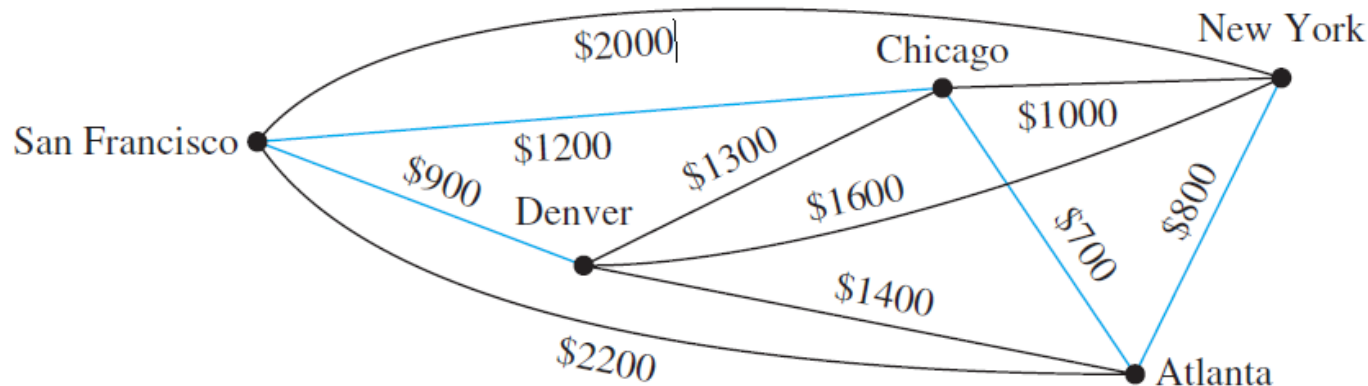
# Árboles

El siguiente grafo indica los costos de una red de comunicaciones. ¿Qué enlaces se deben mantener para asegurar que hay una forma de comunicar cada dos ciudades a un costo mínimo?



# Árboles

El siguiente grafo indica los costos de una red de comunicaciones. ¿Qué enlaces se deben mantener para asegurar que hay una forma de comunicar cada dos ciudades a un costo mínimo?



Choice	Edge	Cost
1	{Chicago, Atlanta}	\$ 700
2	{Atlanta, New York}	\$ 800
3	{Chicago, San Francisco}	\$1200
4	{San Francisco, Denver}	\$ 900
Total:		\$3600

# Árboles

---

## Algoritmo Prim

- Escoga la arista con menor peso y adiciónela al árbol recubridor
- Seleccione la arista con menor peso que sea incidente con el árbol recubridor y que no cree un circuito. Adiciónela al árbol.
- Repita el proceso hasta cuando el árbol tenga  $n-1$  aristas ( $n$  es el número de vértices)

# Árboles

---

## Algoritmo Prim

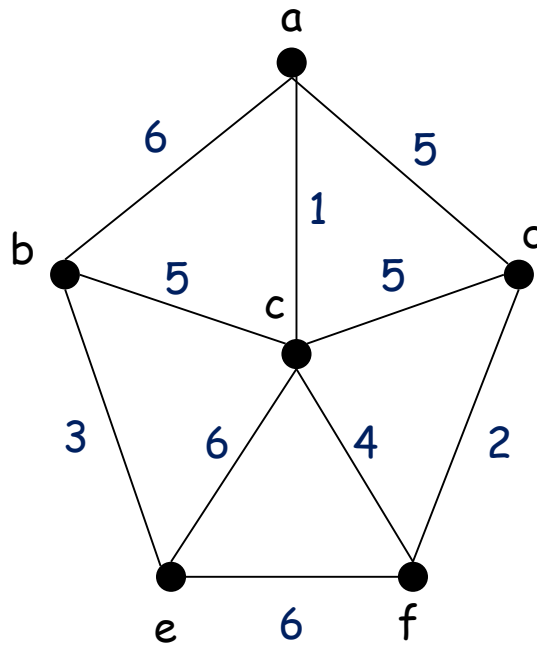
- Escoga la arista con menor peso y adiciónela al árbol recubridor
- Seleccione la arista con menor peso que sea incidente con el árbol recubridor y que no cree un circuito. Adiciónela al árbol.
- Repita el proceso hasta cuando el árbol tenga  $n-1$  aristas ( $n$  es el número de vértices)

Al usar el algoritmo de Prim se pueden obtener árboles recubridores diferentes para un mismo grafo

# Árboles

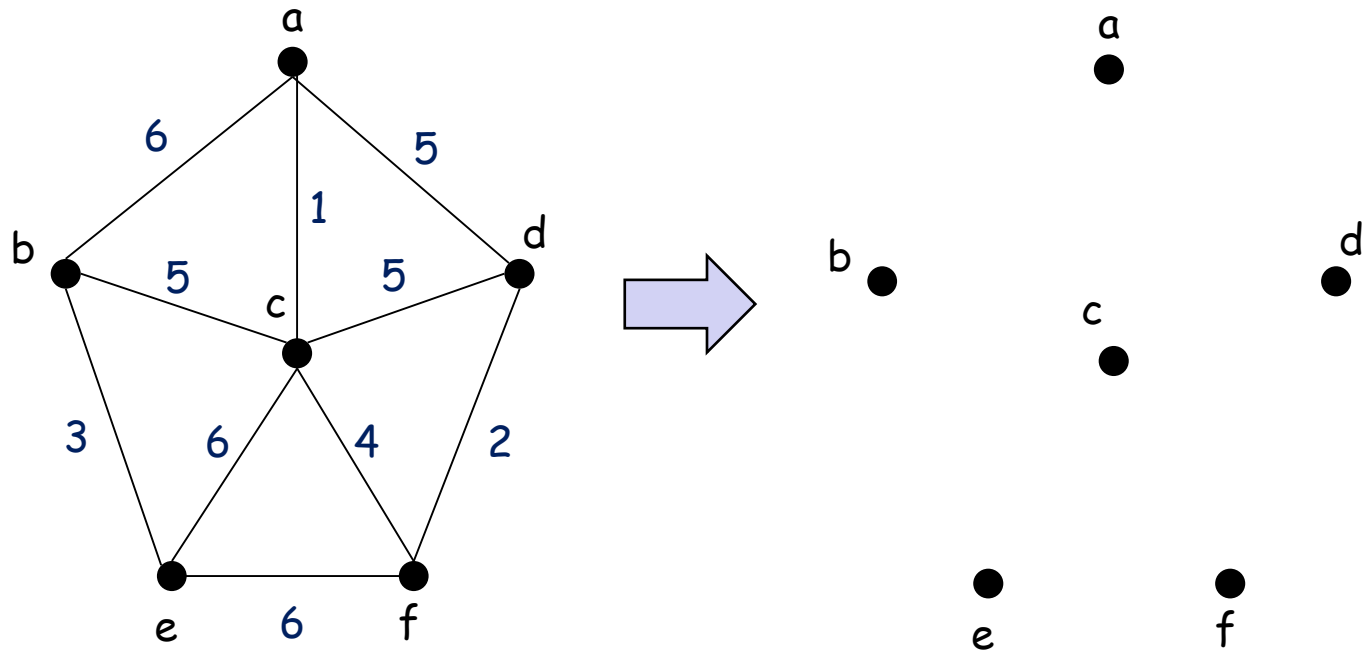
---

Encontrar un árbol recubridor mínimo usando el algoritmo de Prim



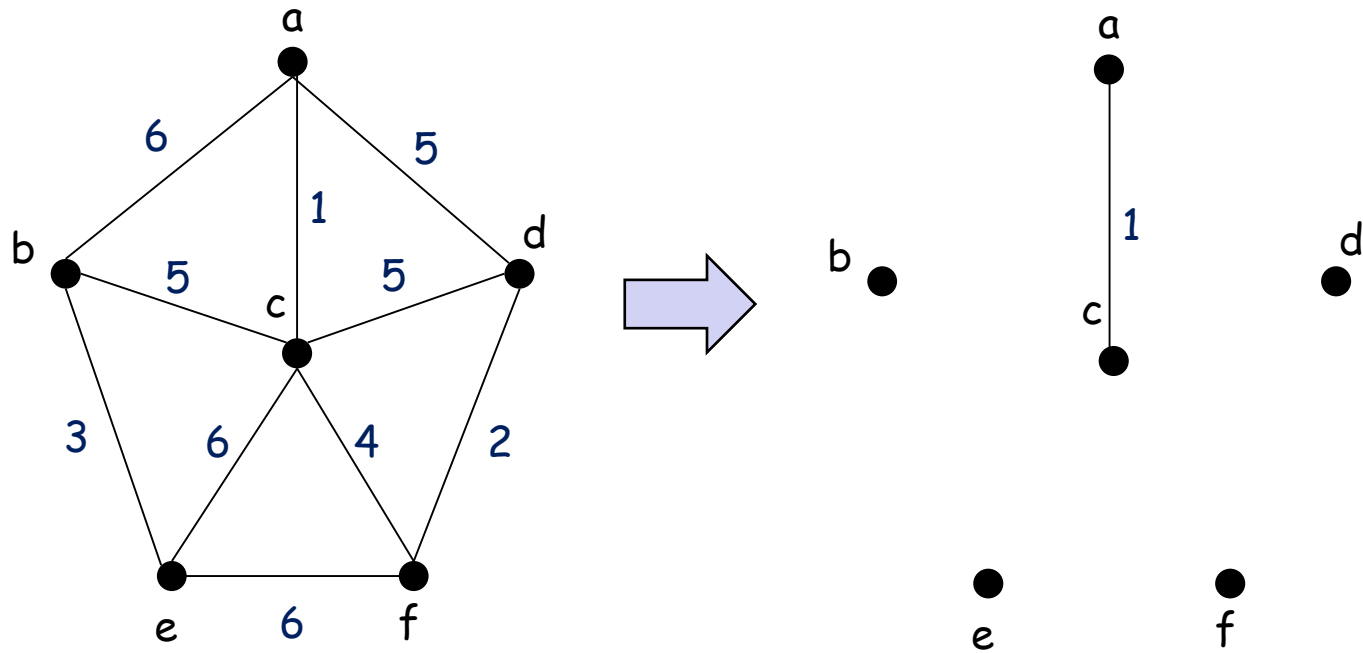
# Árboles

---



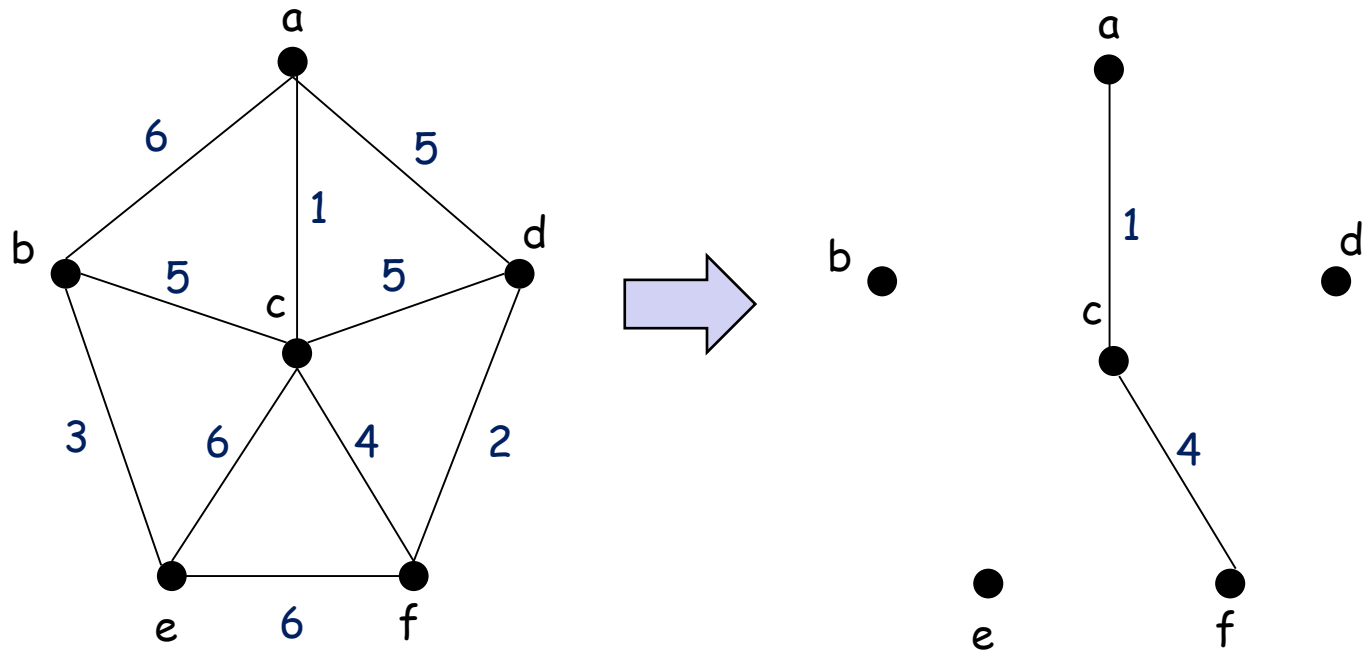
# Árboles

---



# Árboles

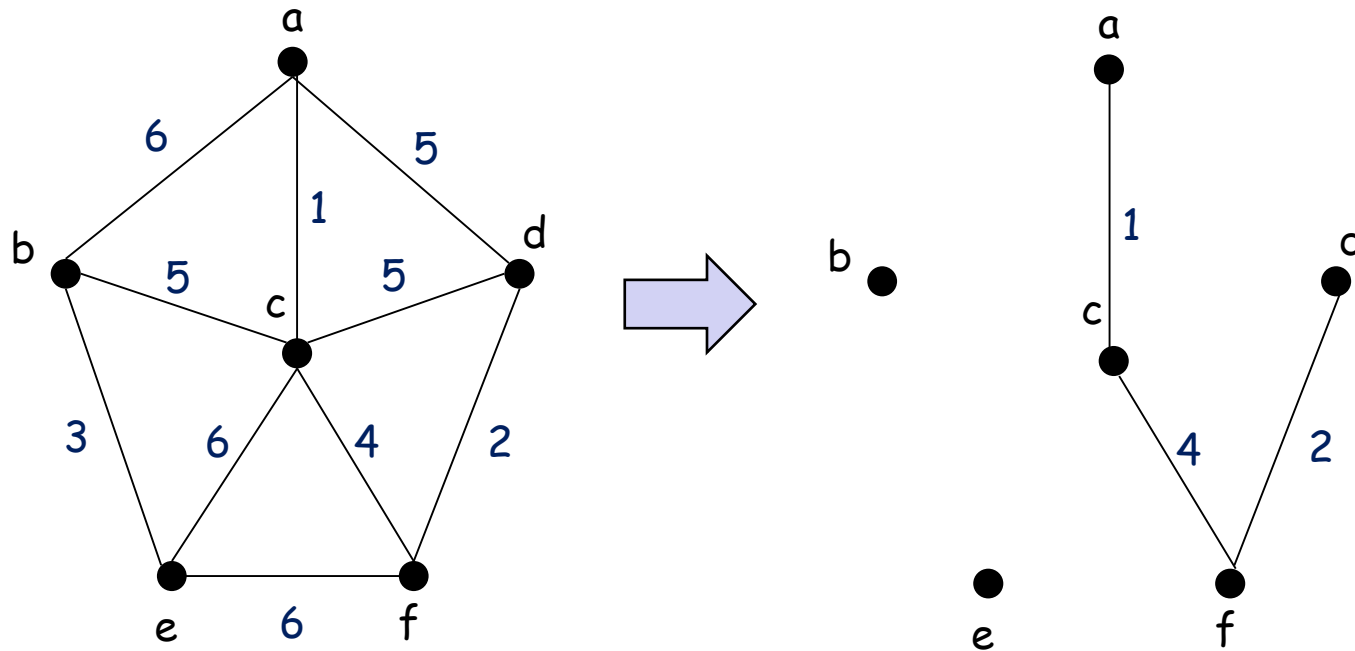
---





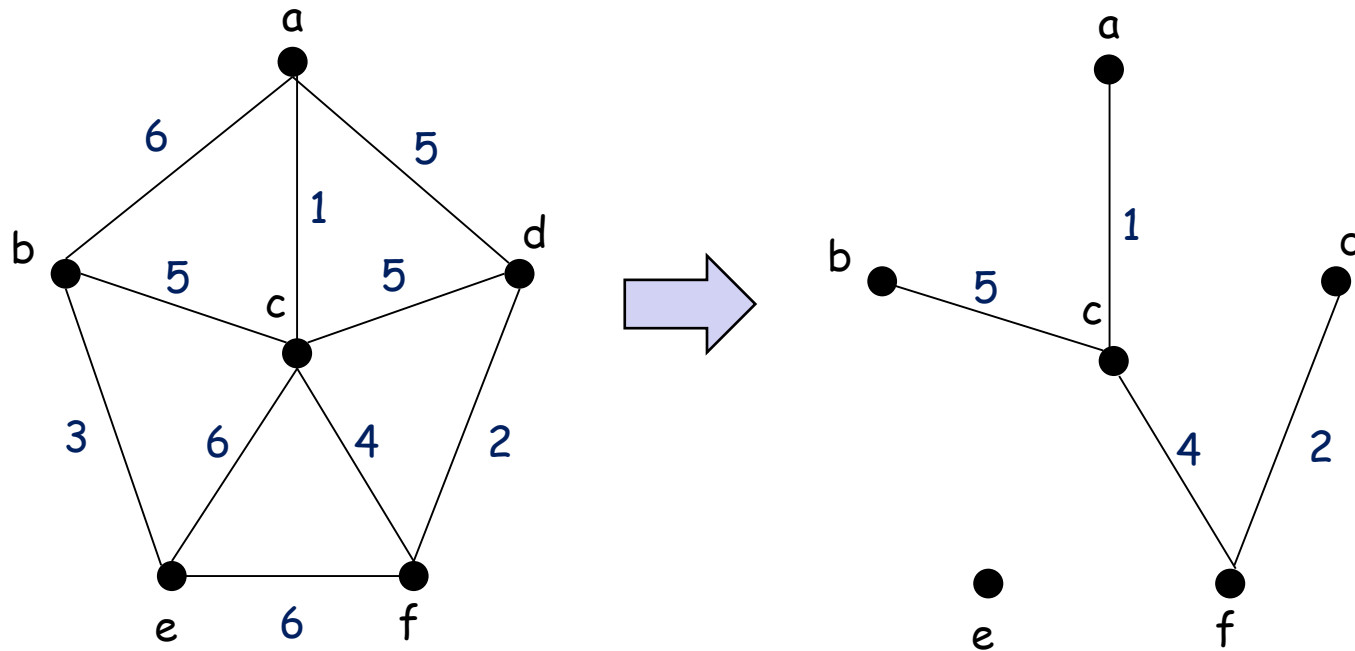
# Árboles

---



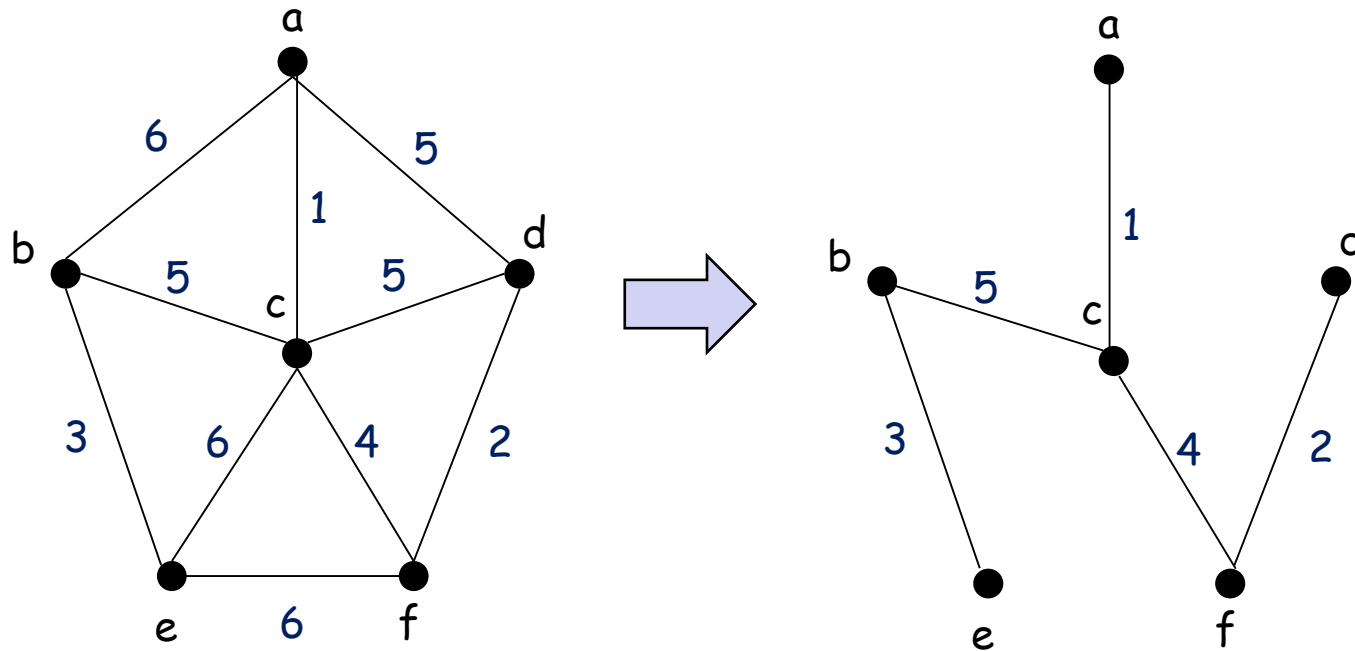
# Árboles

---



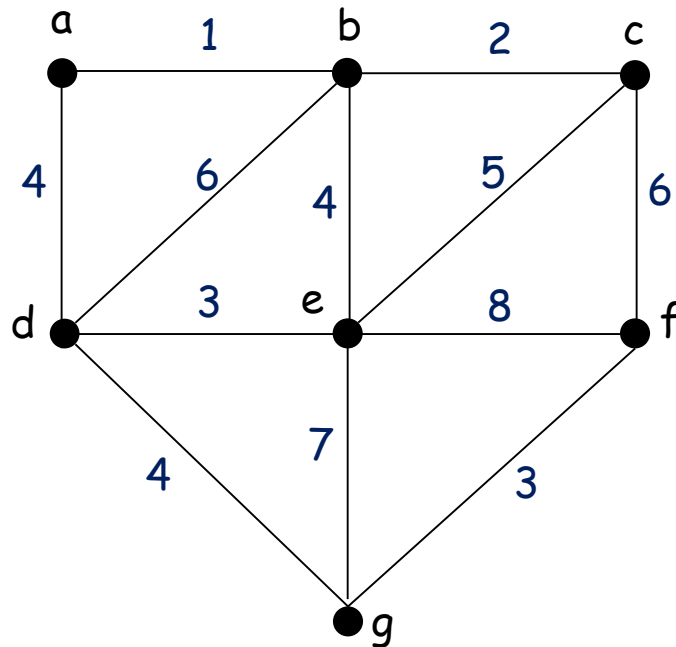
# Árboles

---

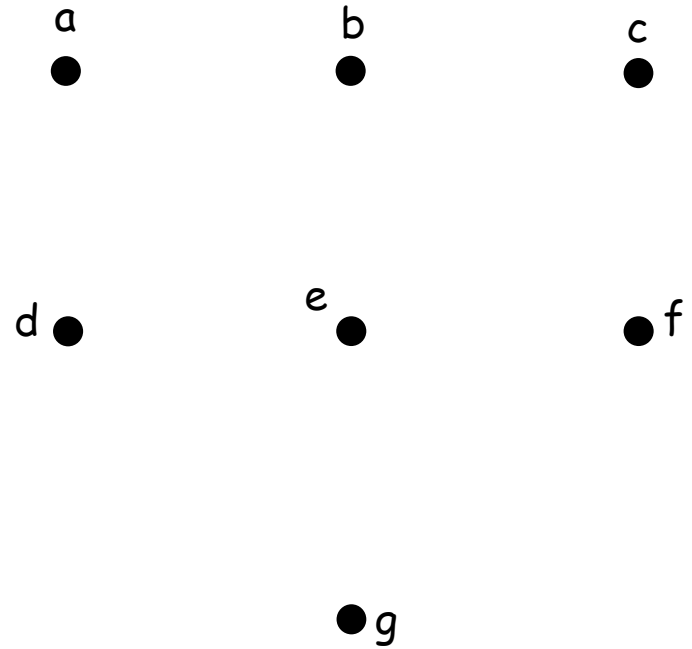
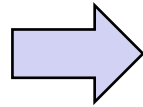
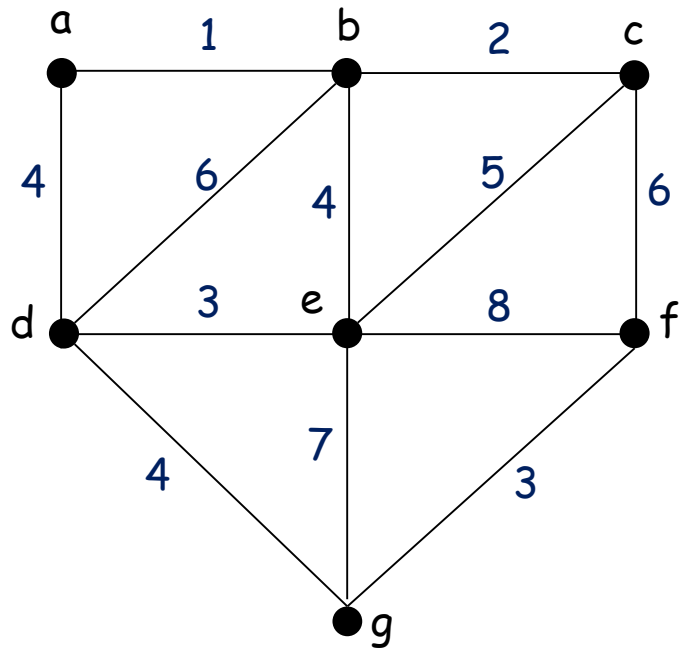


# Árboles

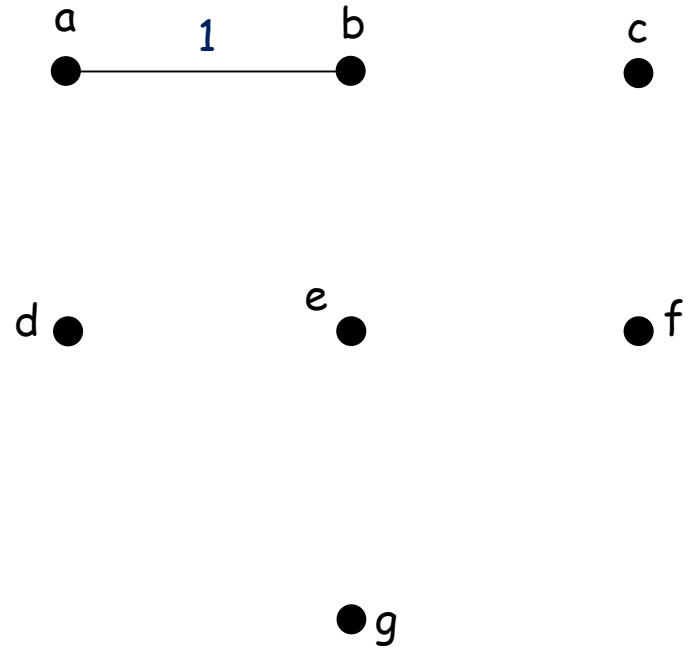
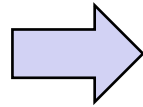
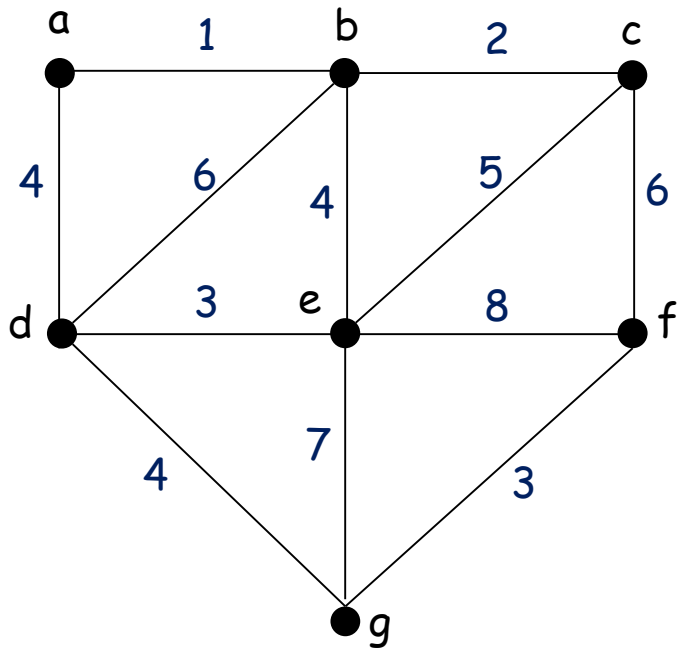
Encontrar un árbol recubridor mínimo usando el algoritmo de Prim



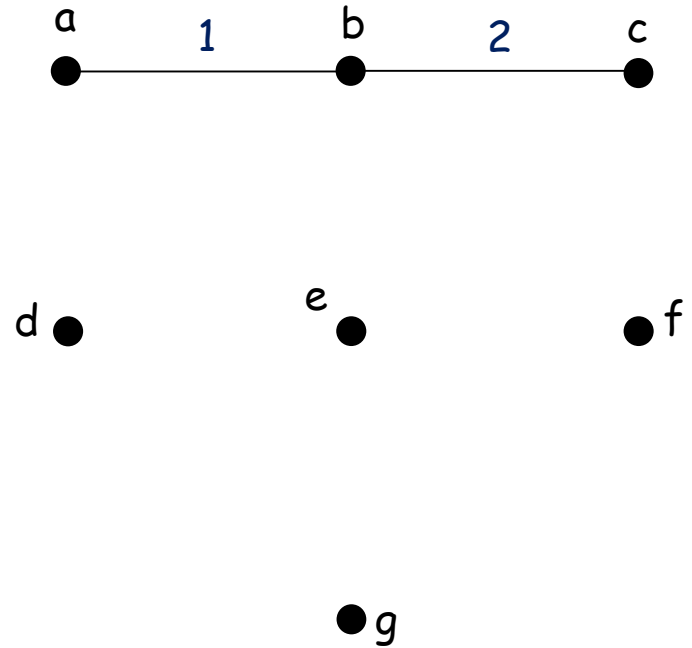
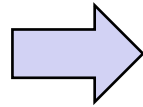
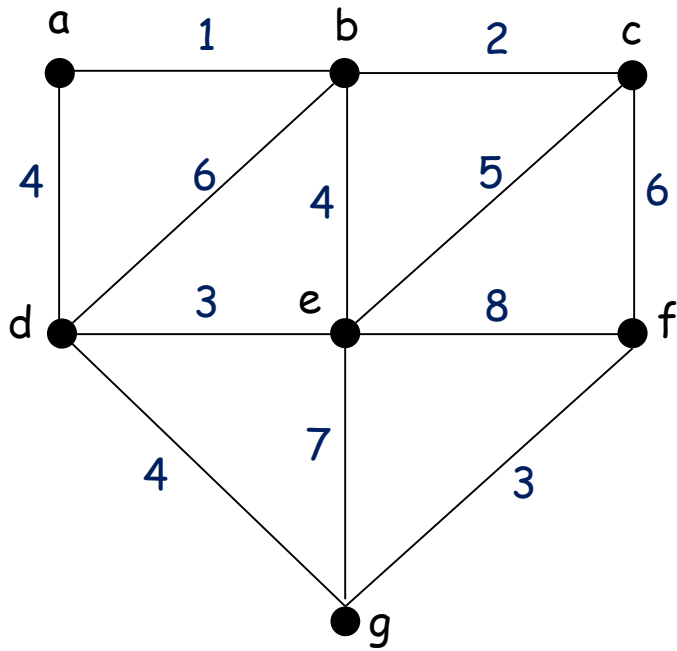
# Árboles



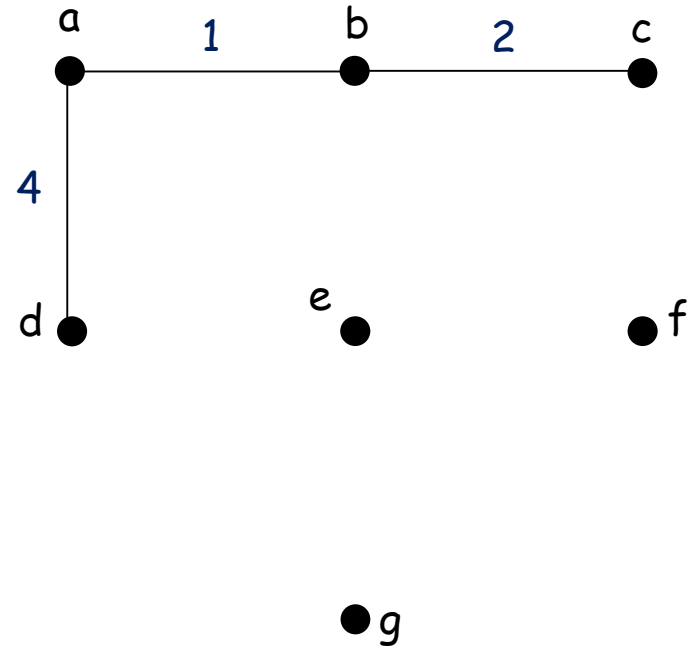
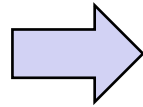
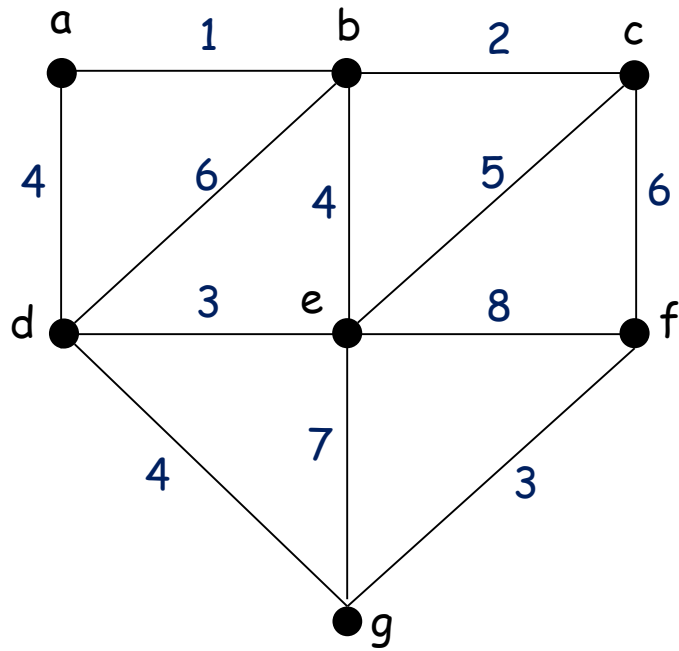
# Árboles



# Árboles

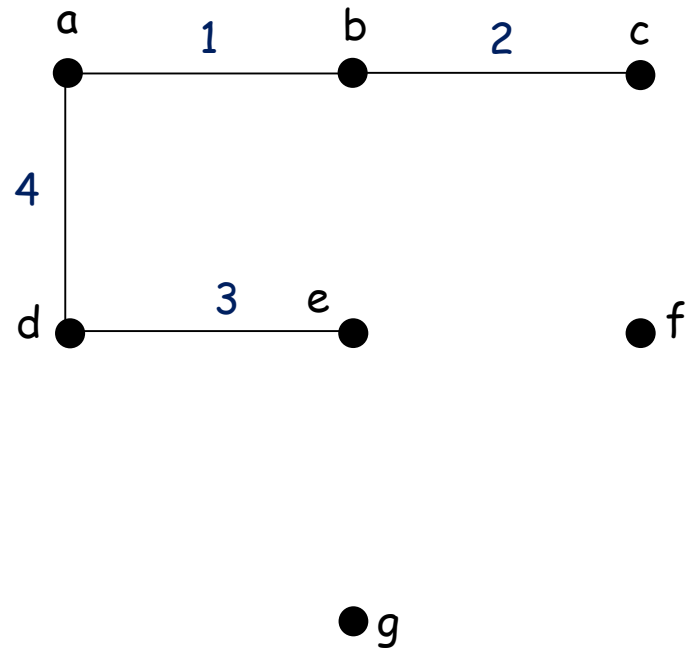
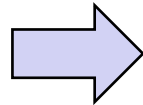
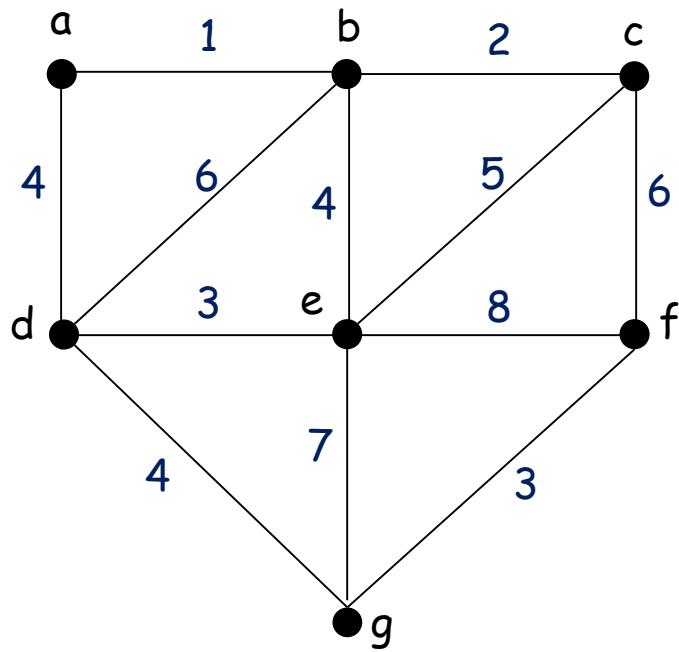


# Árboles

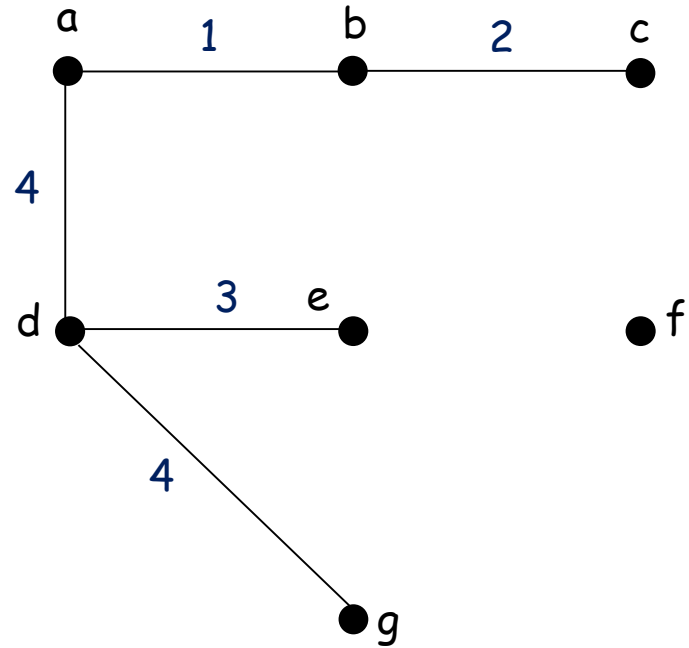
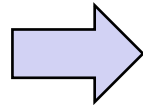
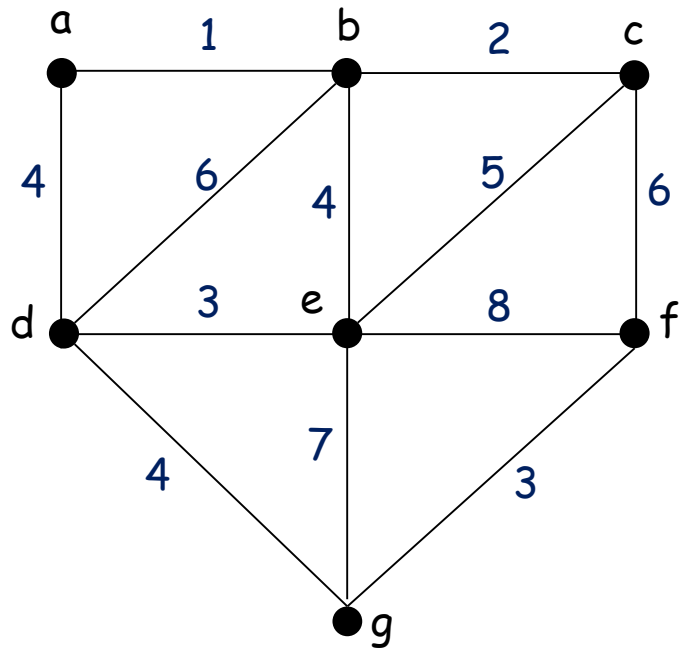




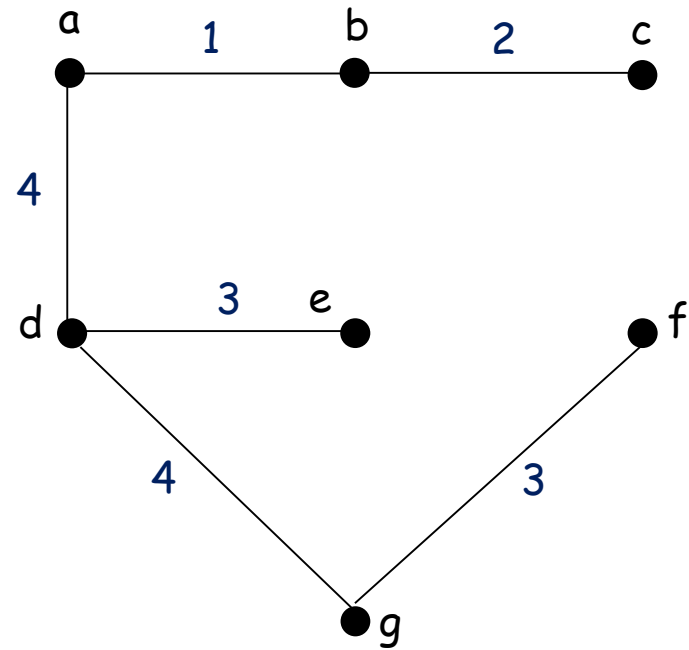
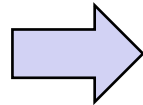
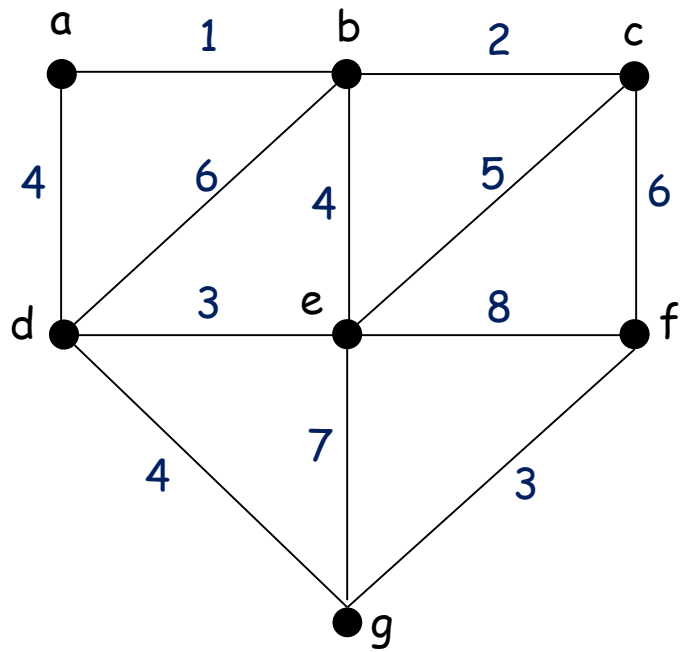
# Árboles



# Árboles

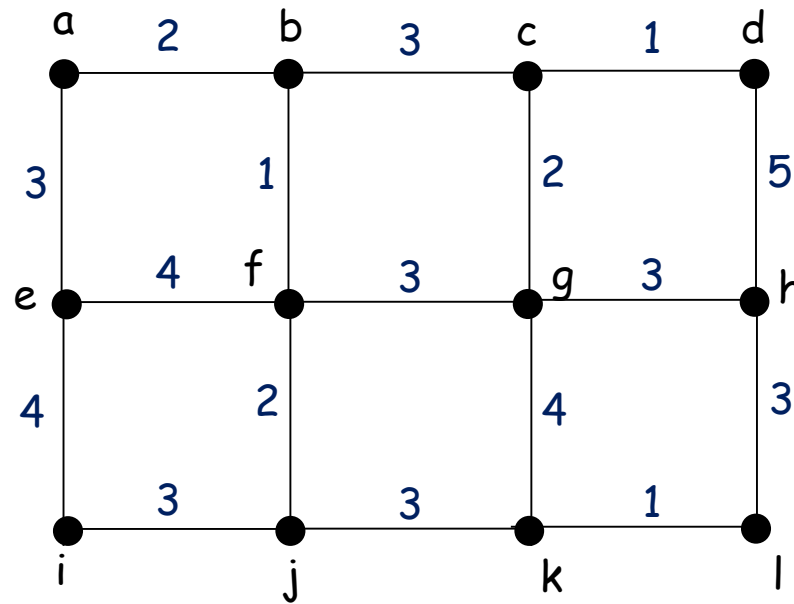


# Árboles

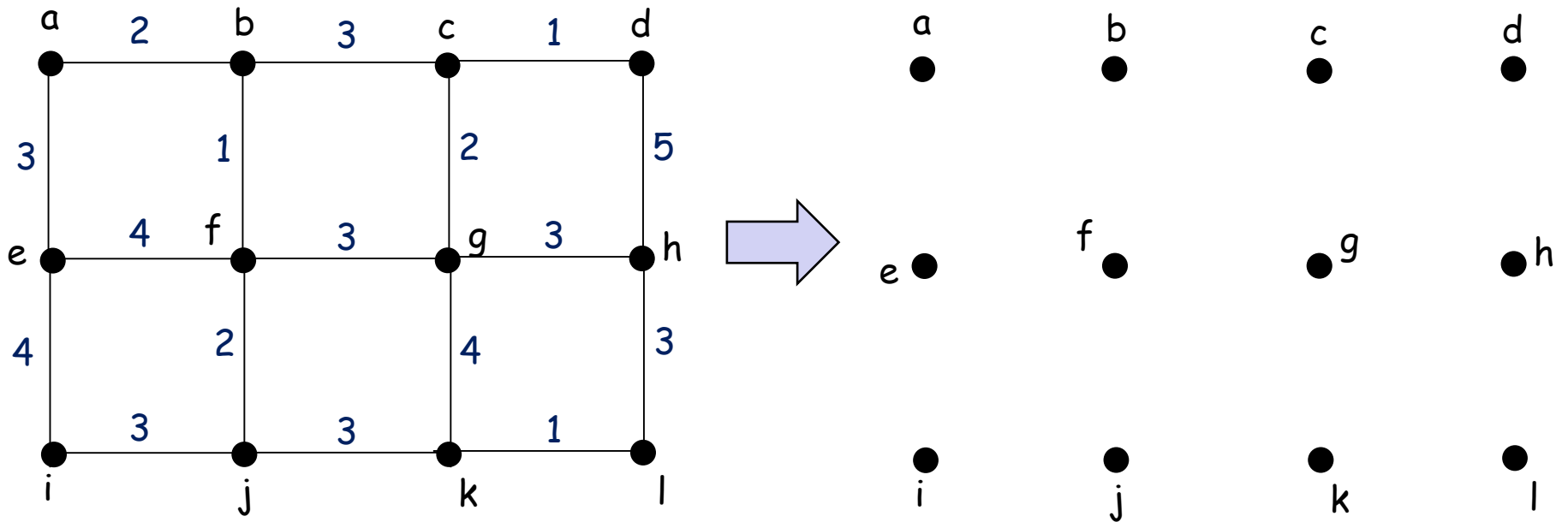


# Árboles

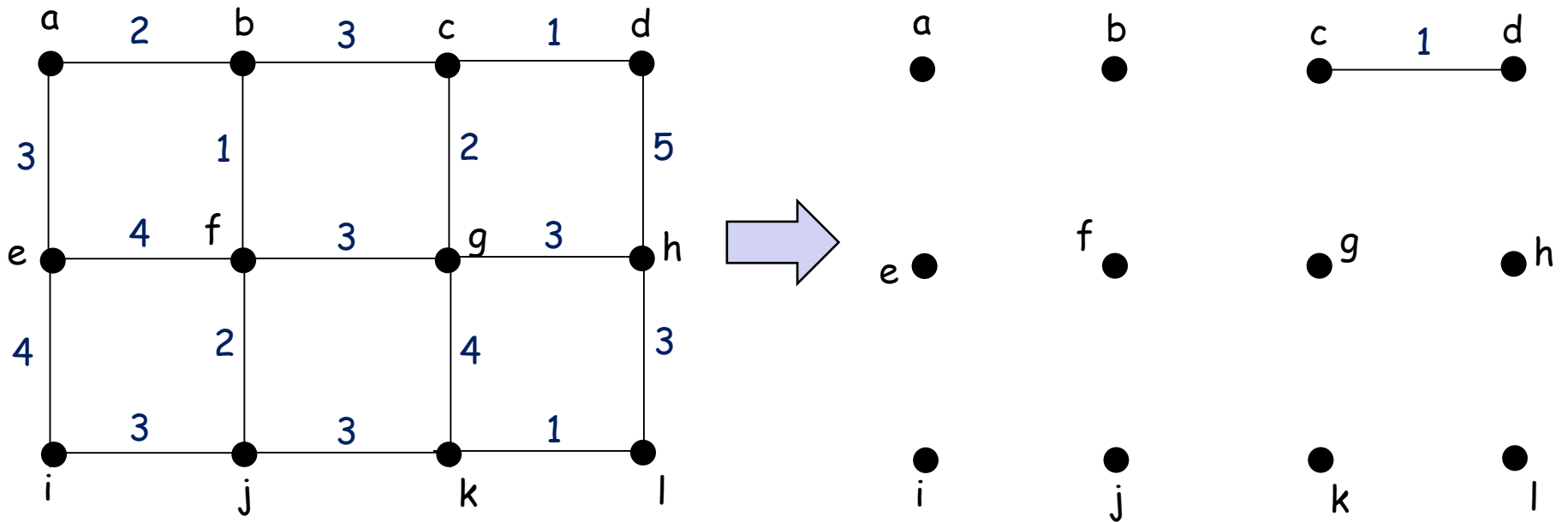
Encontrar un árbol recubridor mínimo usando el algoritmo de Prim



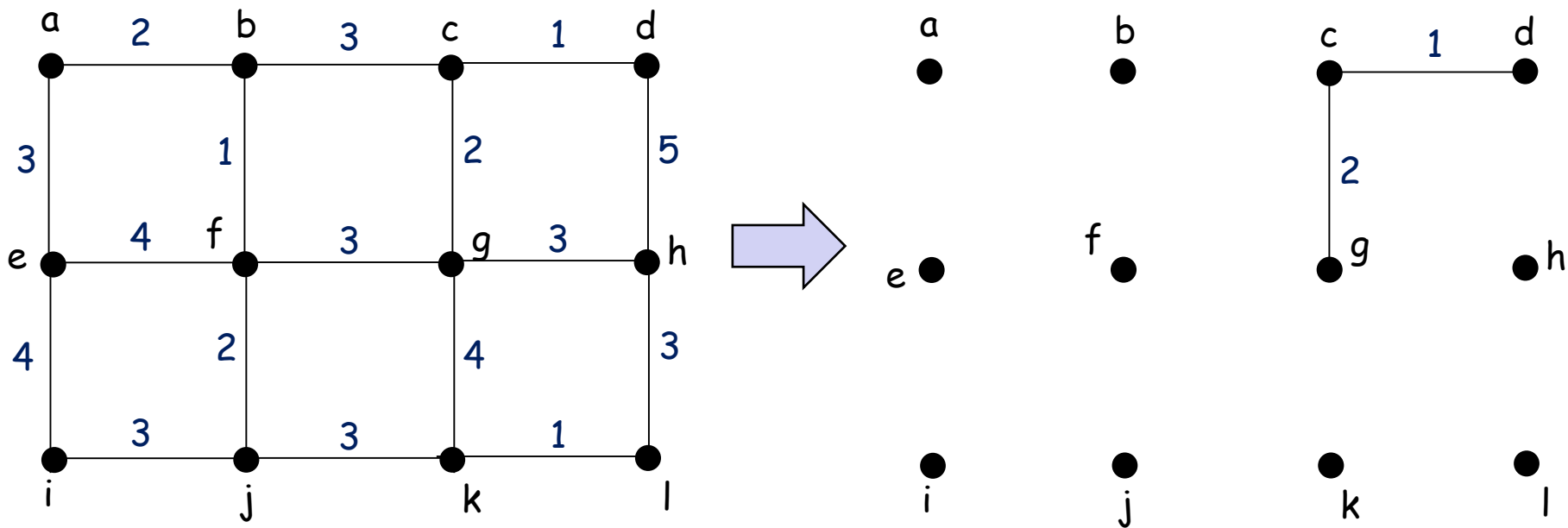
# Árboles



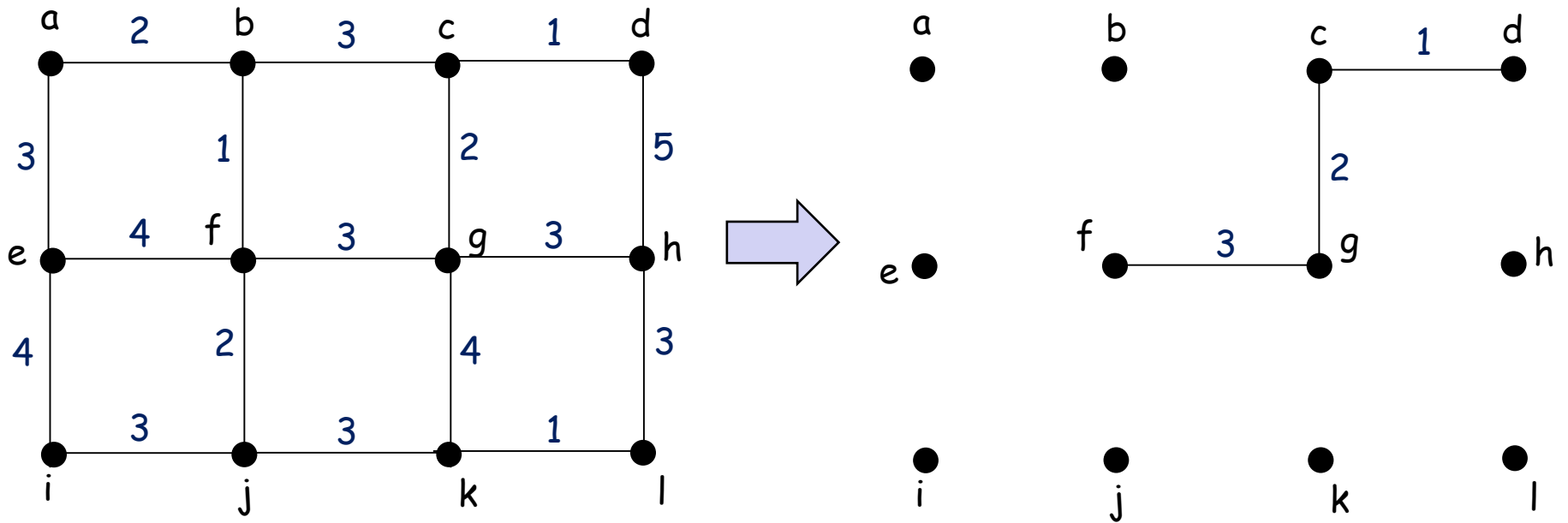
# Árboles



# Árboles

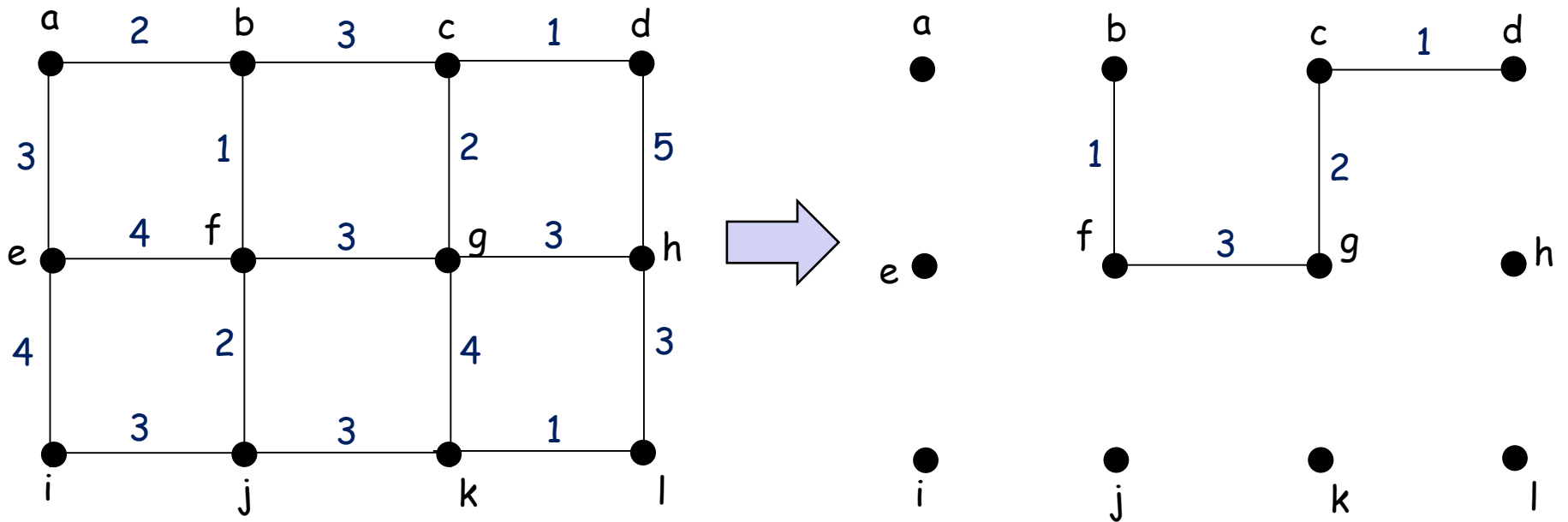


# Árboles

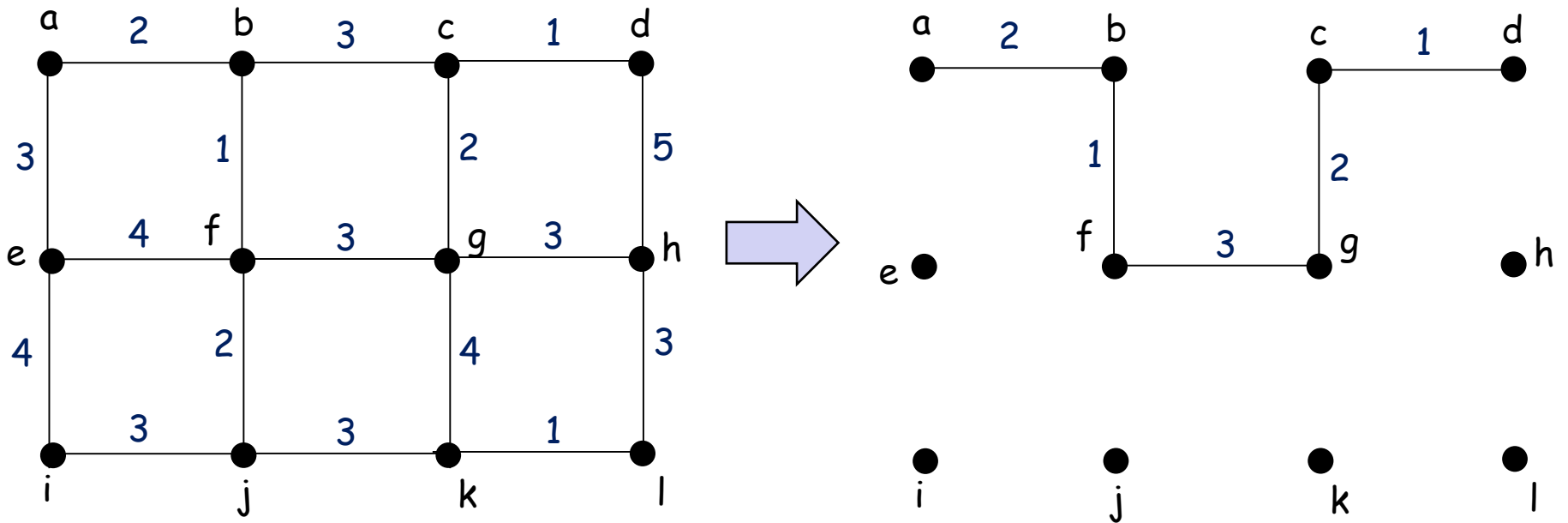




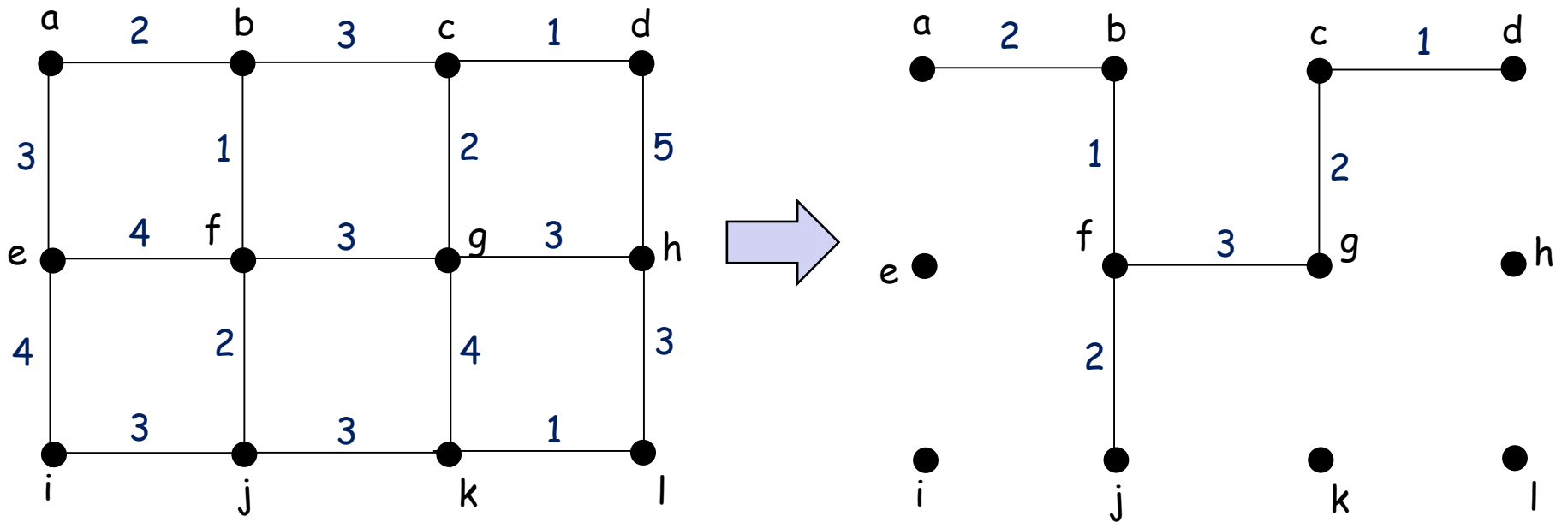
# Árboles



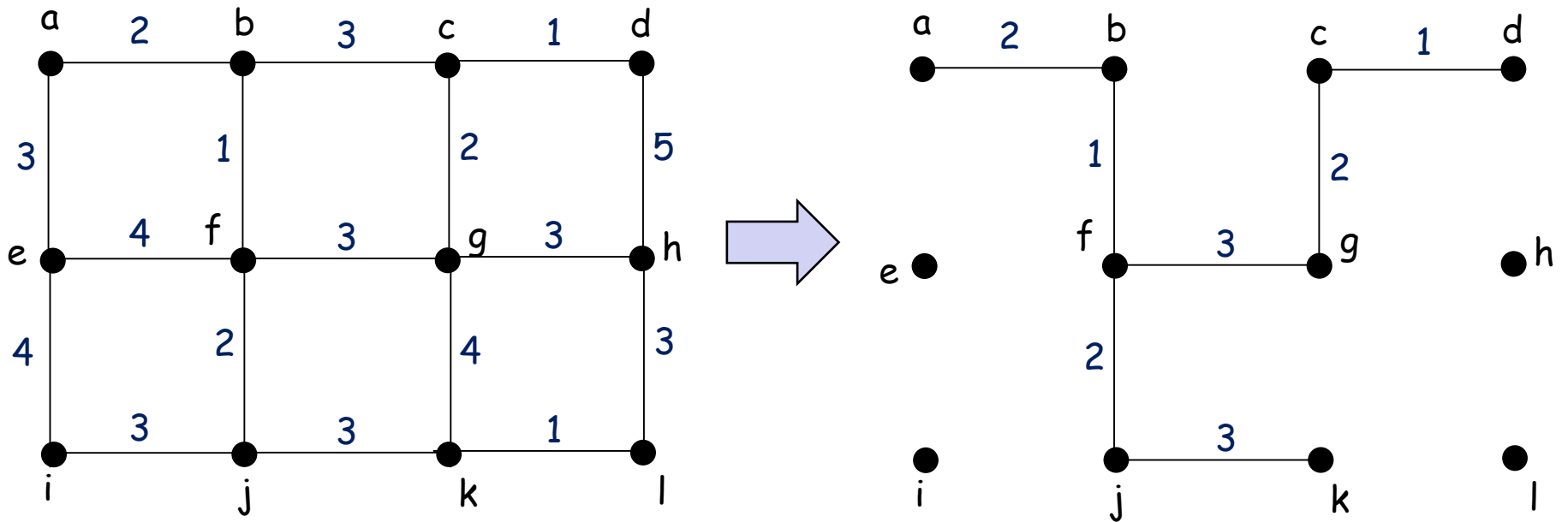
# Árboles



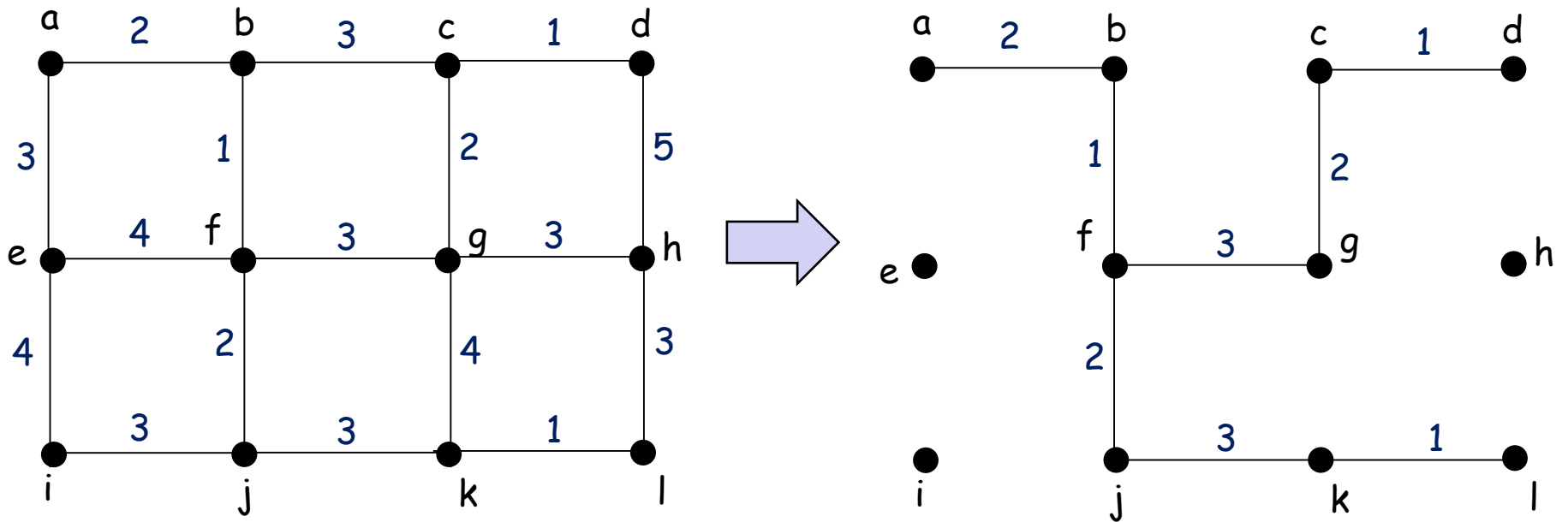
# Árboles



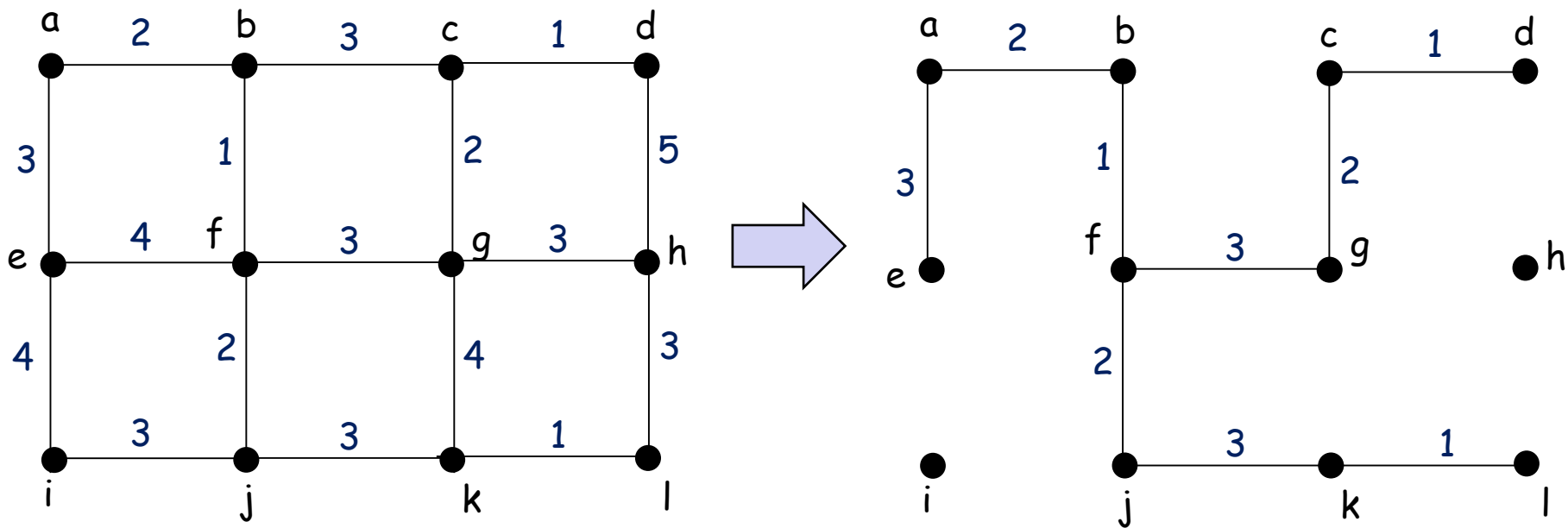
# Árboles



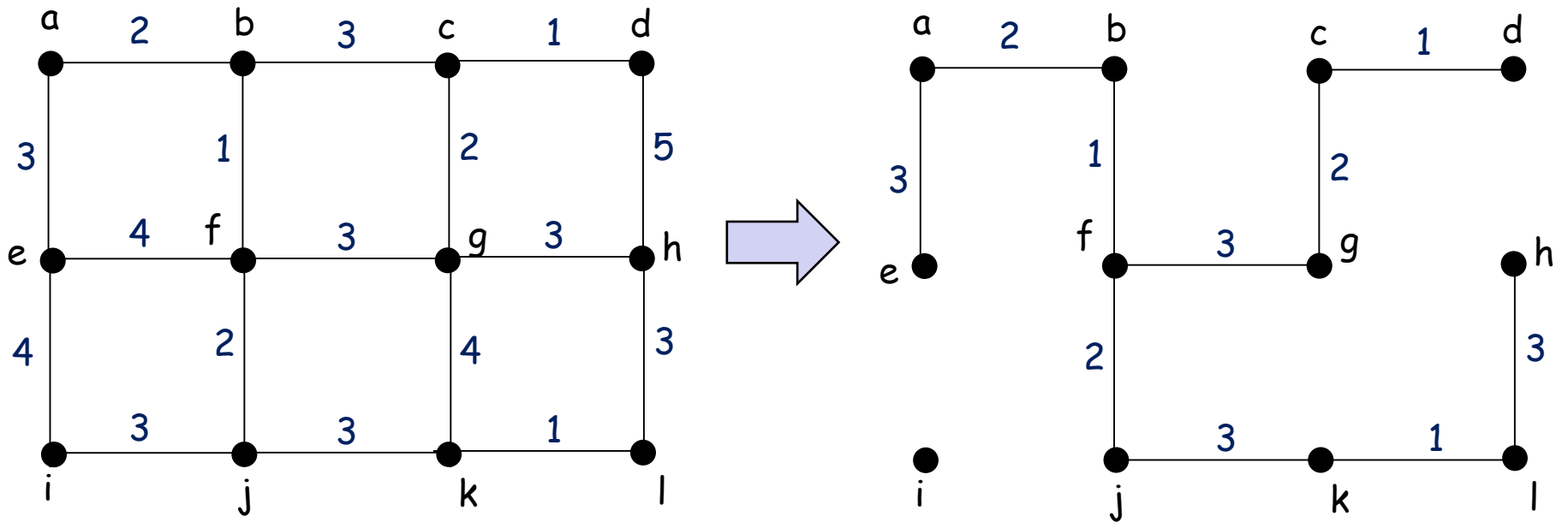
# Árboles



# Árboles

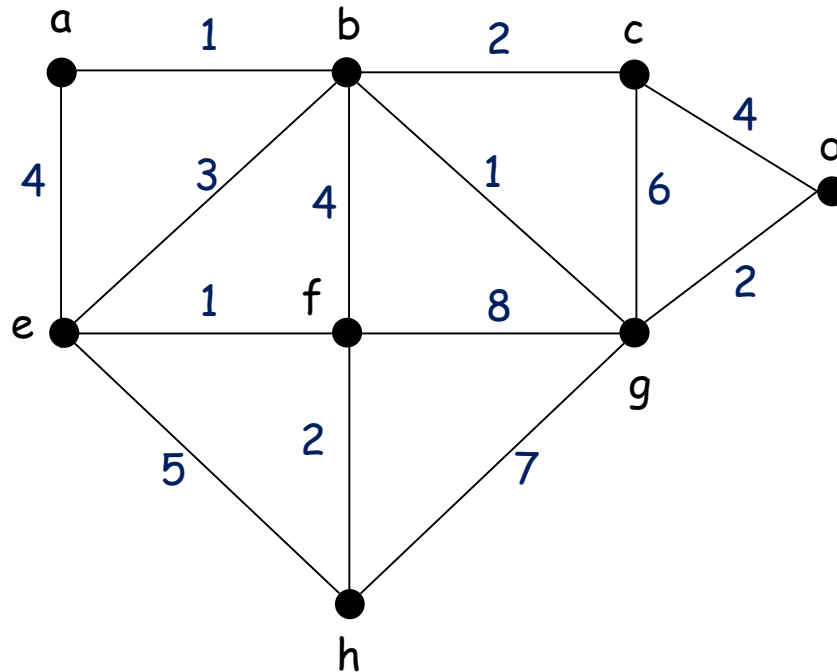


# Árboles



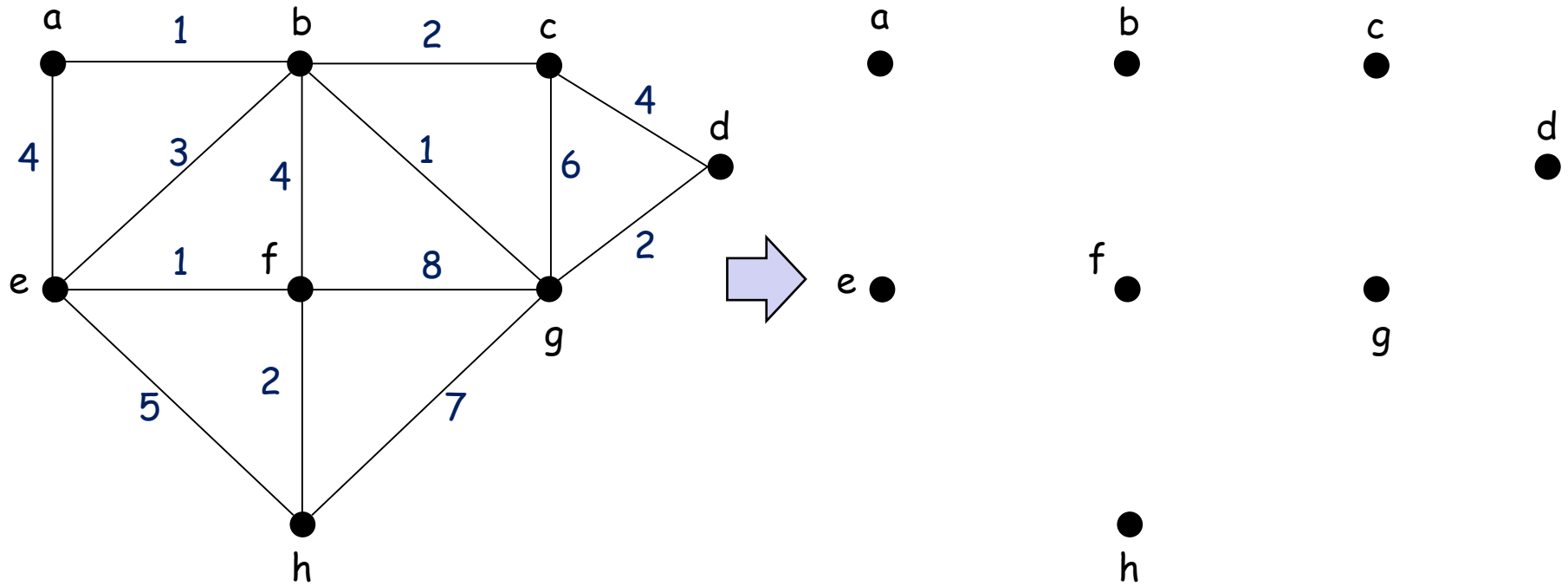
# Árboles

Encontrar un árbol recubridor mínimo usando el algoritmo de Prim

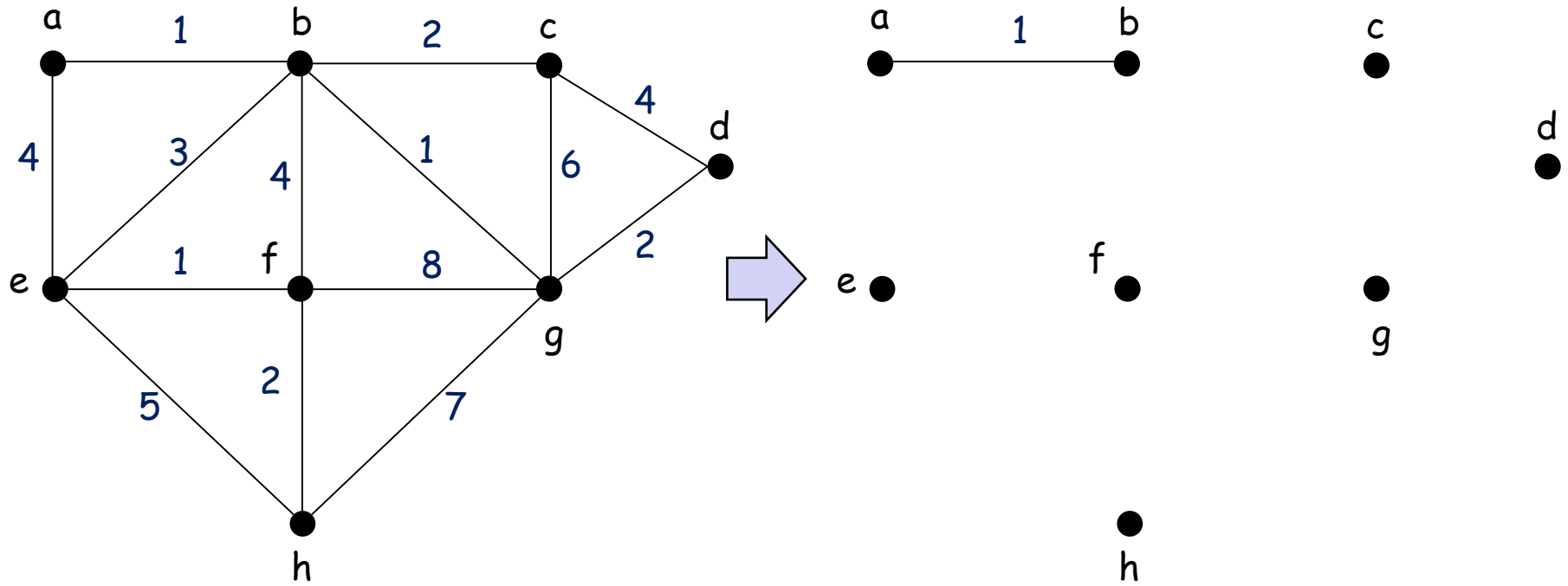




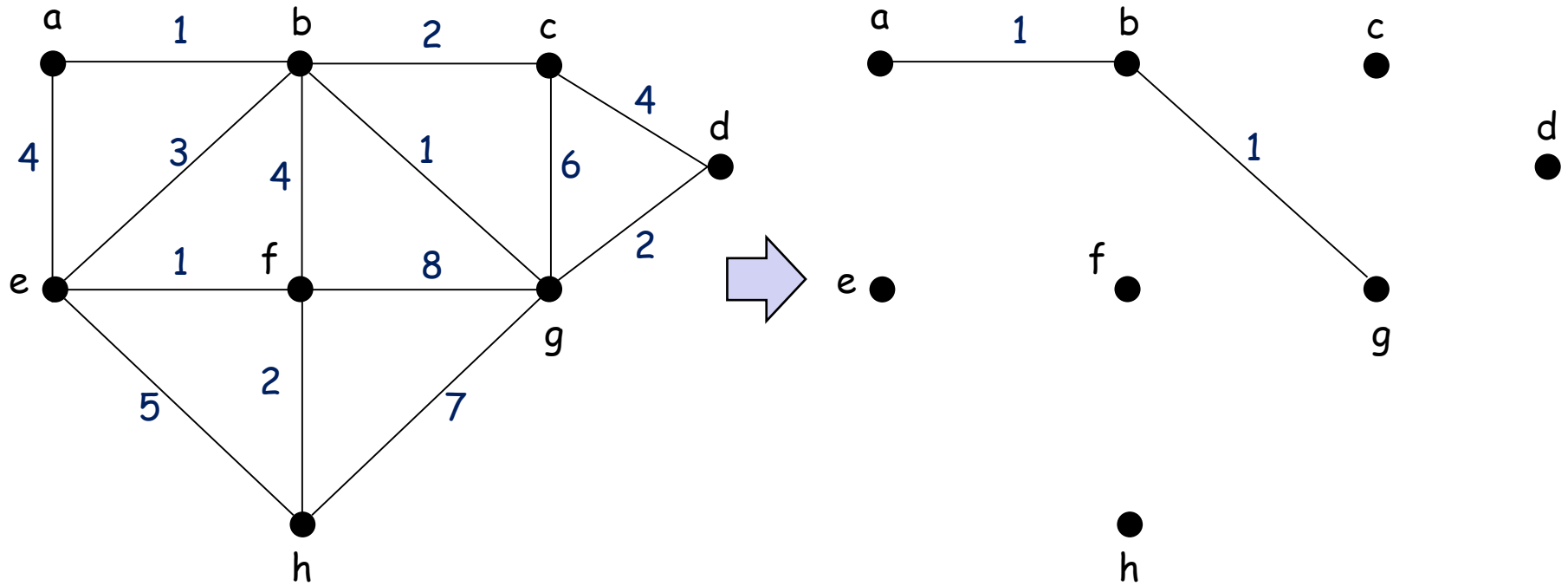
# Árboles



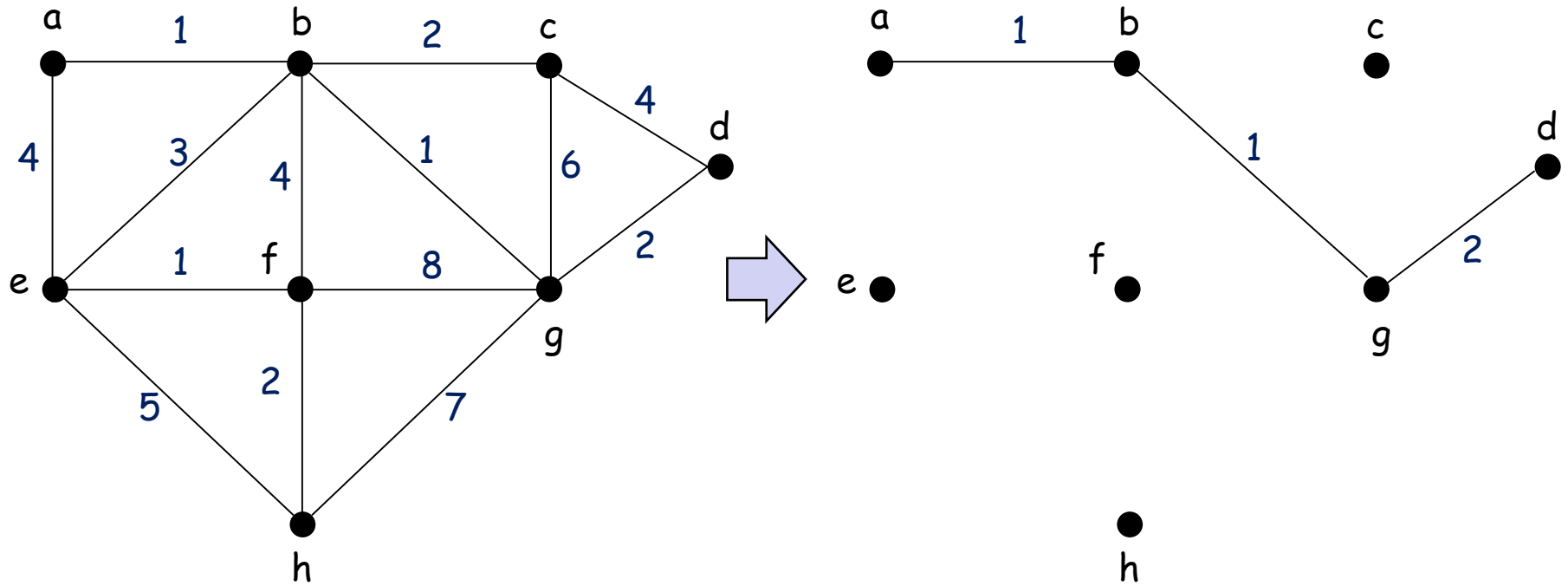
# Árboles



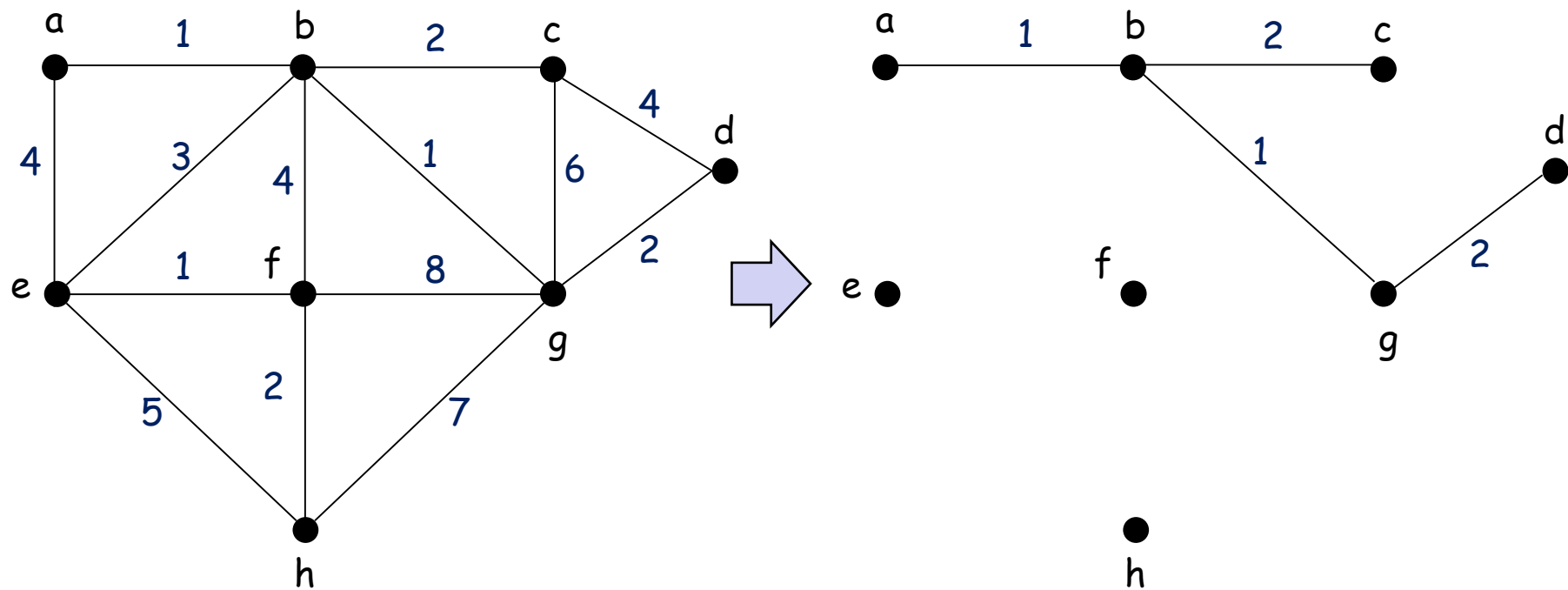
# Árboles



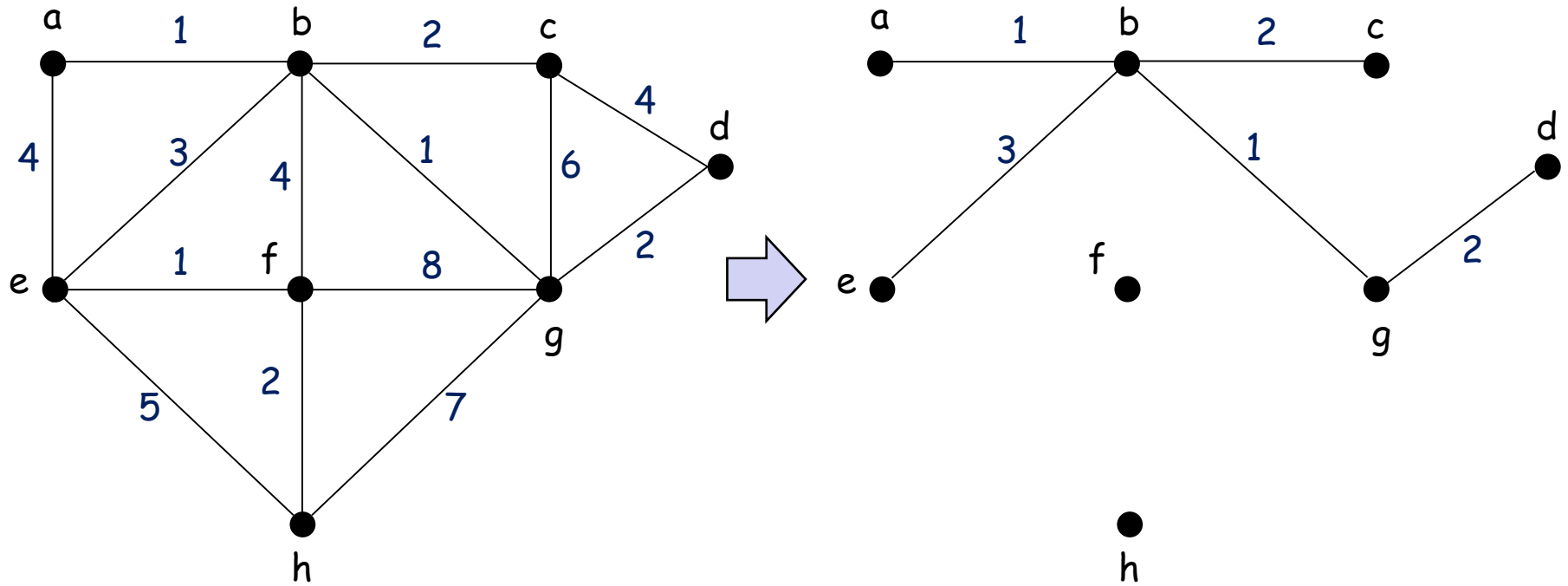
# Árboles



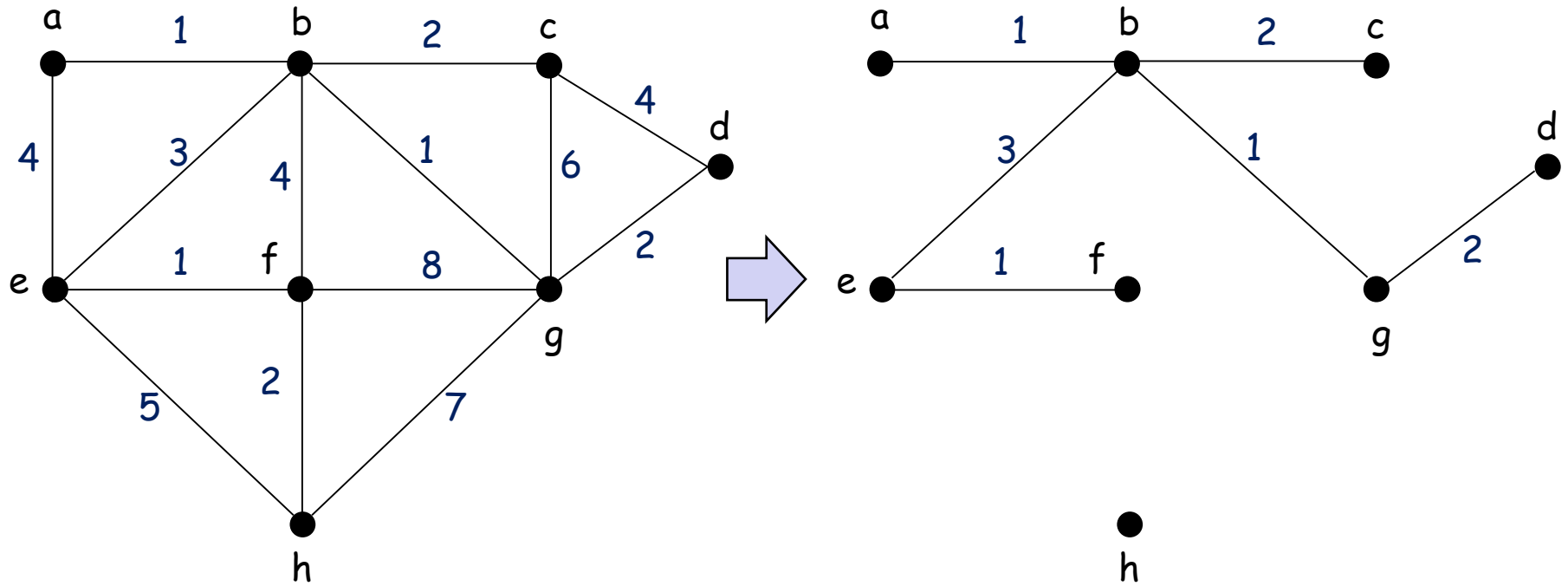
# Árboles



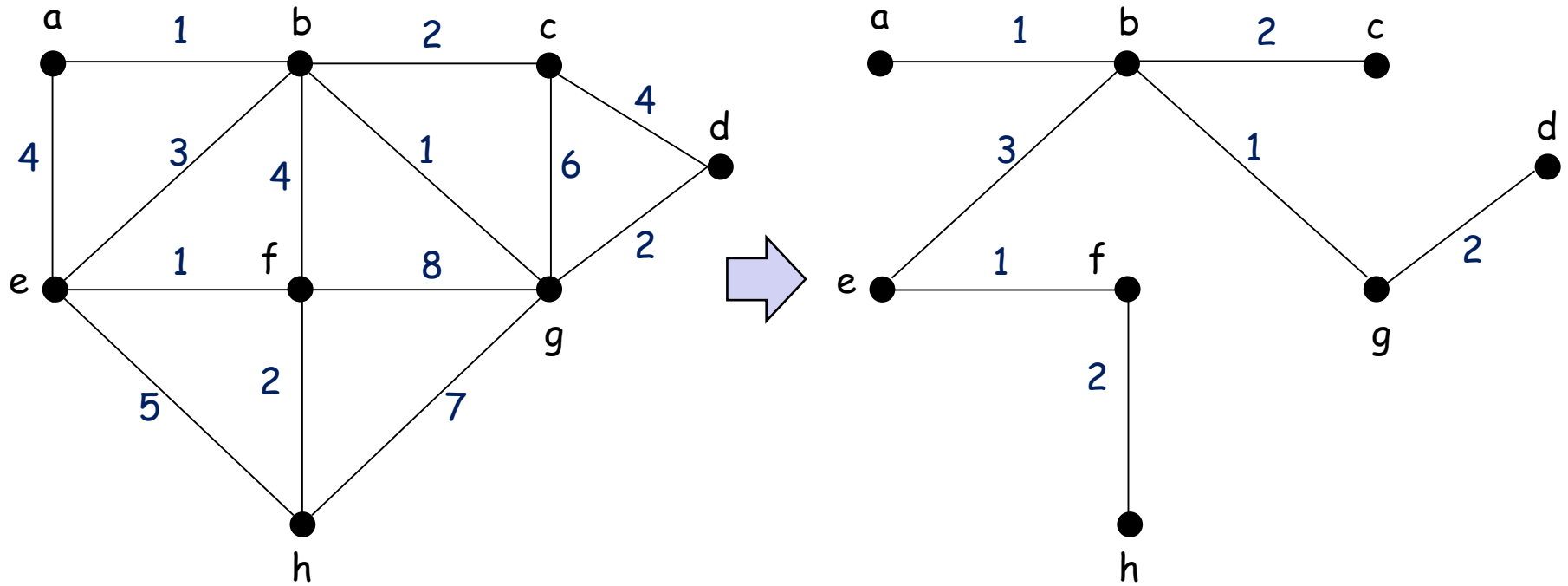
# Árboles



# Árboles



# Árboles





# Árboles

---

## Algoritmo de Kruskal

- Seleccione la arista con menor peso y adiciónela al árbol recubridor
- Adicione al árbol la arista con menor peso que no cree un circuito
- Repita el proceso cuando el árbol tenga  $n-1$  aristas ( $n$  es el número de vértices)

# Árboles

---

## Algoritmo de Kruskal

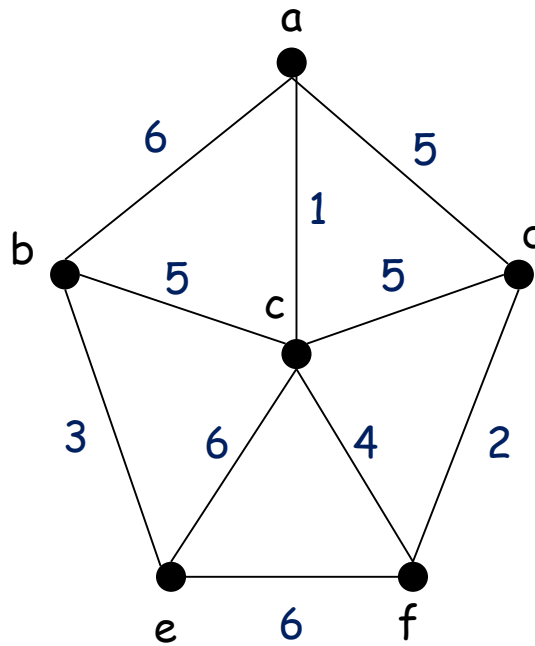
- Seleccione la arista con menor peso y adiciónela al árbol recubridor
- Adicione al árbol la arista con menor peso que no cree un circuito
- Repita el proceso cuando el árbol tenga  $n-1$  aristas ( $n$  es el número de vértices)

La arista que se selecciona con el algoritmo de Prim debe ser incidente en el árbol recubridor, mientras que en el algoritmo de Kruskal no

# Árboles

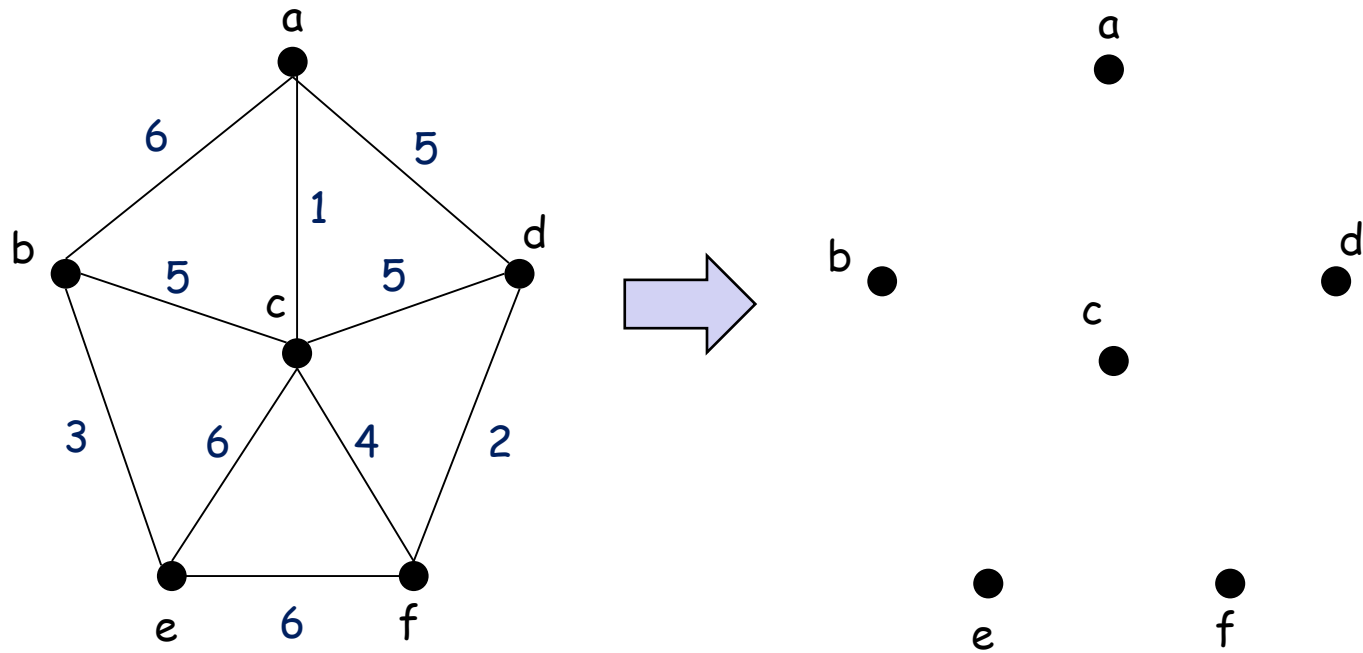
---

Encontrar un árbol recubridor mínimo usando el algoritmo de Kruskal



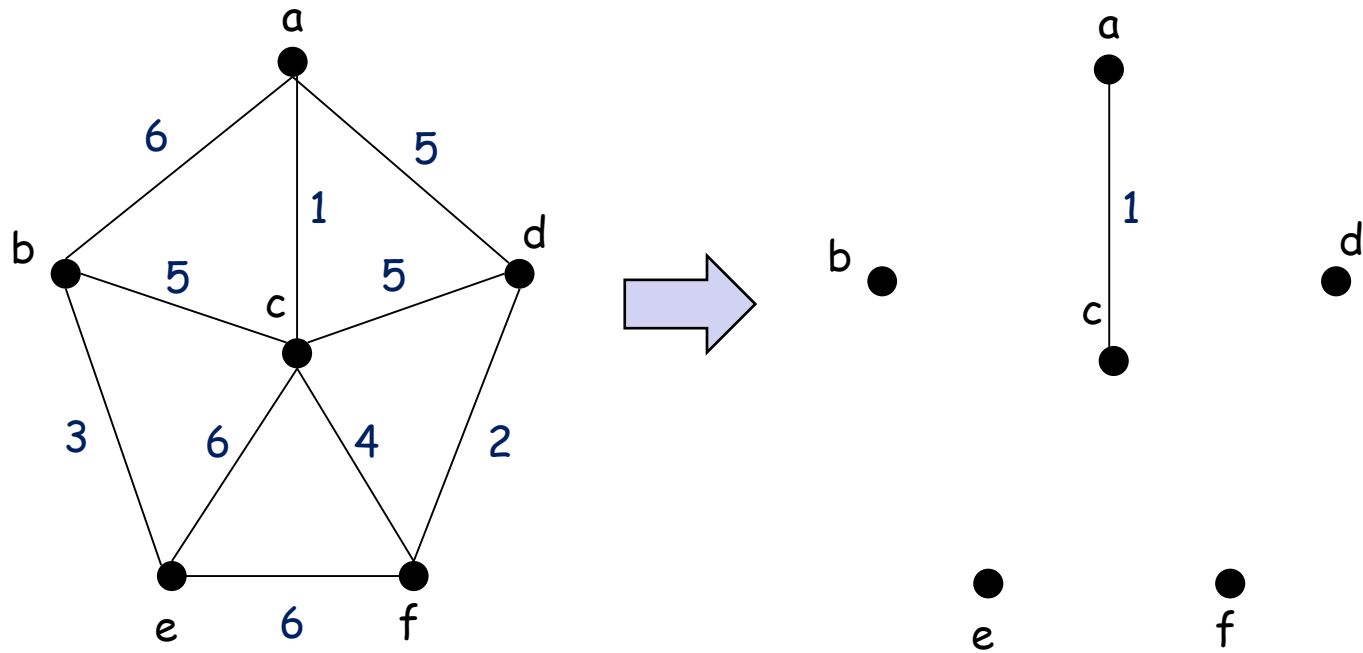
# Árboles

---



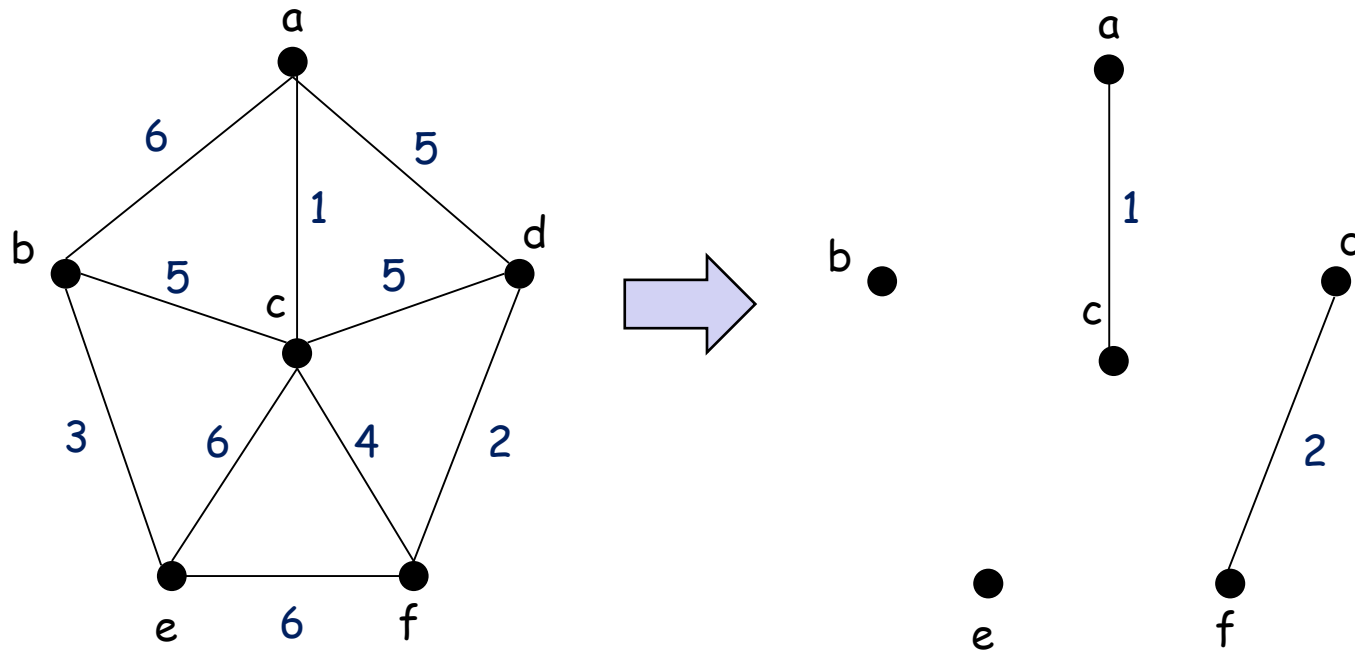
# Árboles

---



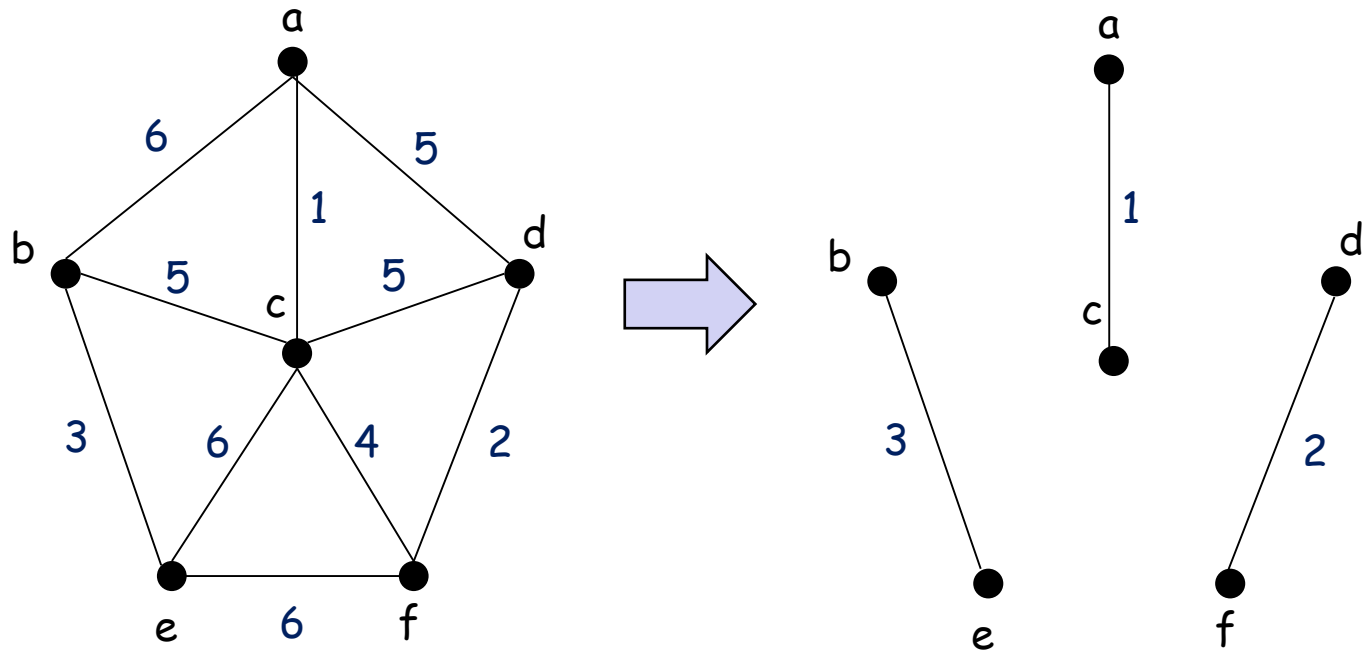
# Árboles

---



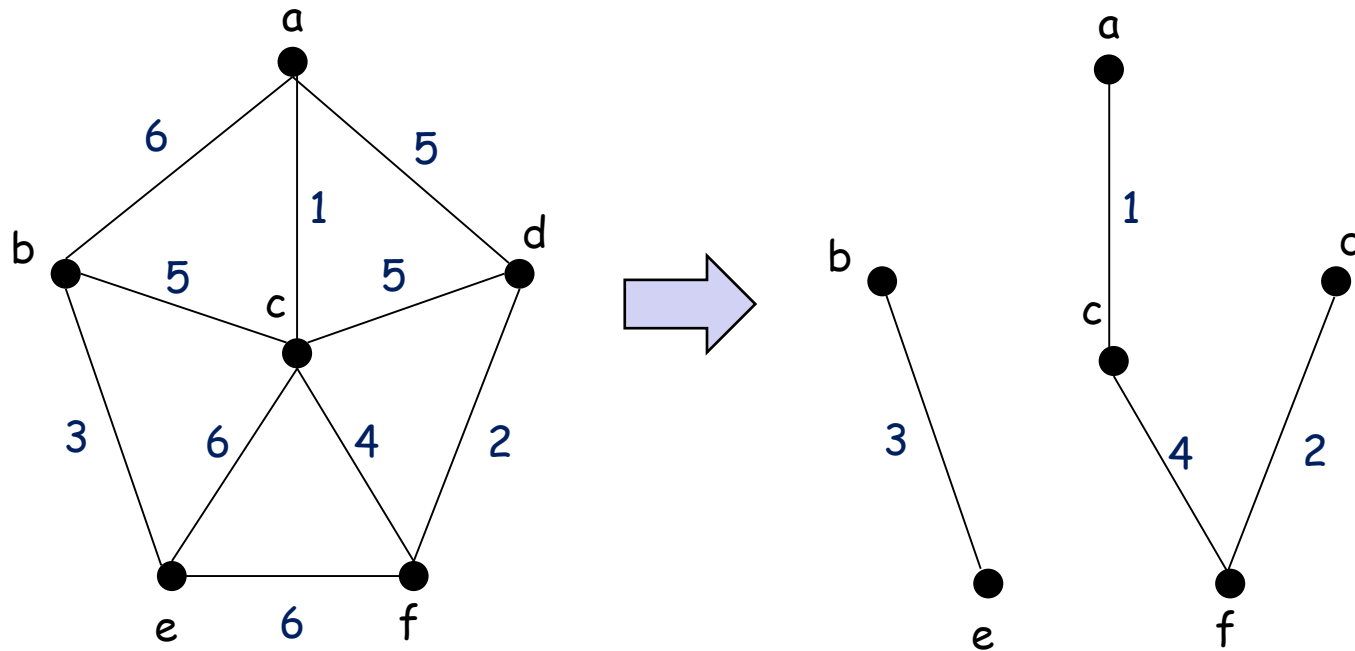
# Árboles

---



# Árboles

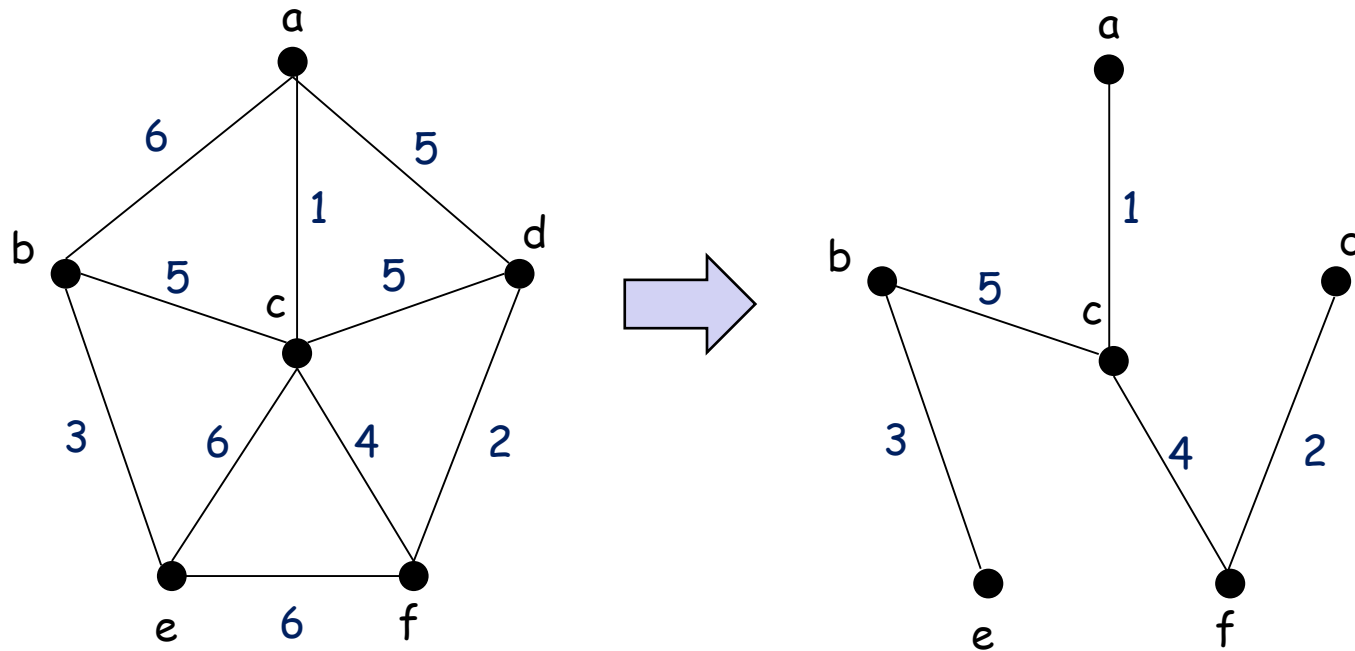
---





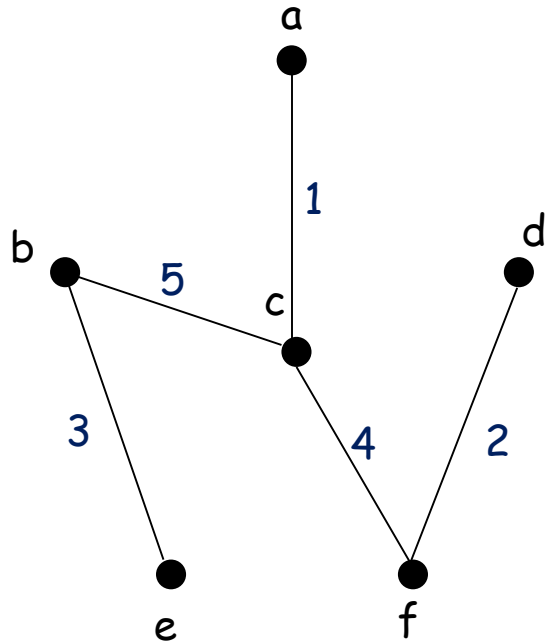
# Árboles

---

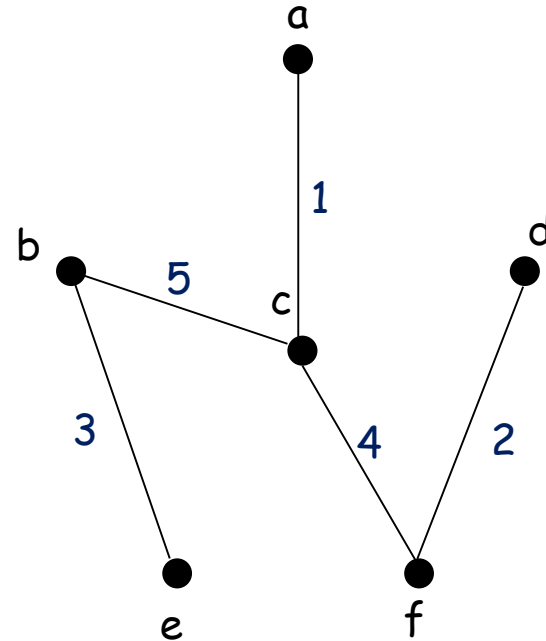


# Árboles

---



Árbol recubridor  
mínimo obtenido con el  
algoritmo de Prim

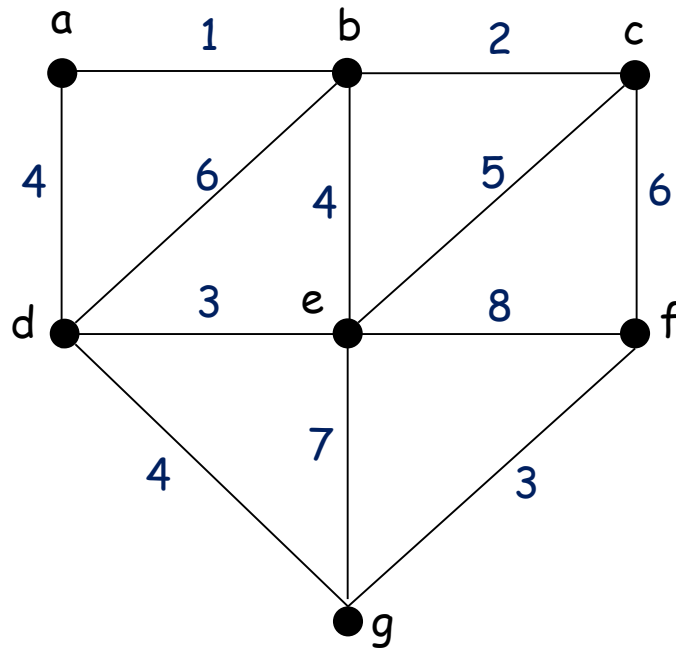


Árbol recubridor  
mínimo obtenido con el  
algoritmo de Kruskal

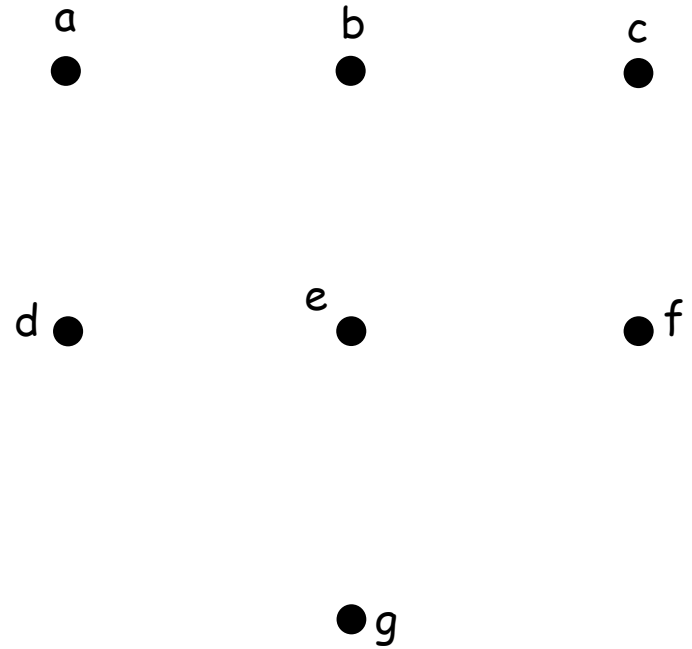
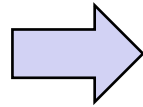
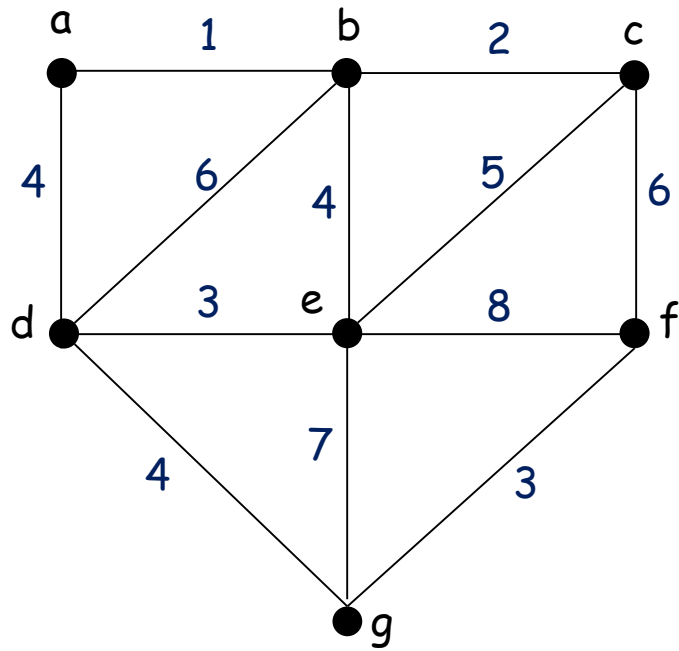
# Árboles

---

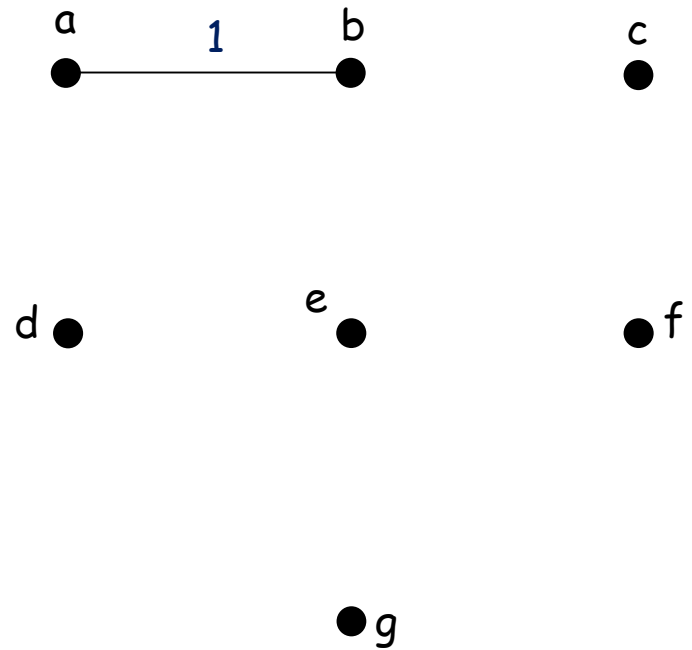
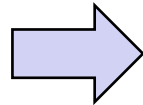
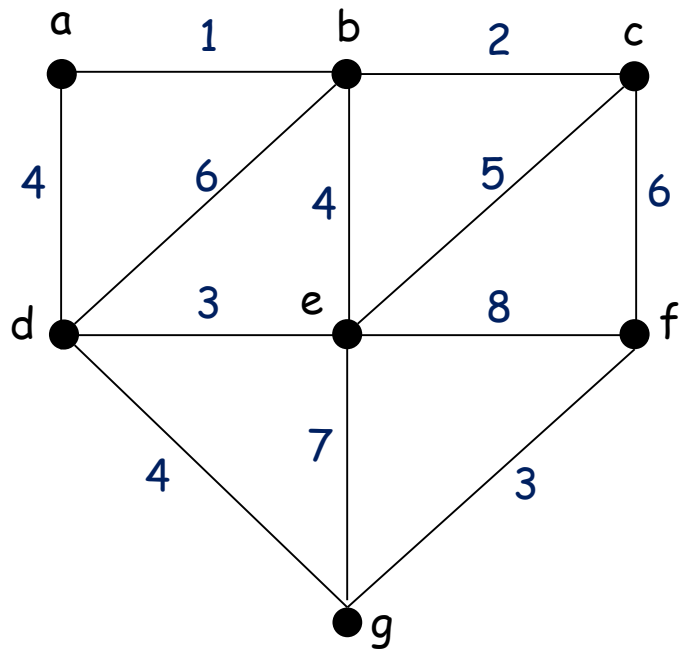
Encontrar un árbol recubridor mínimo usando el algoritmo de Kruskal



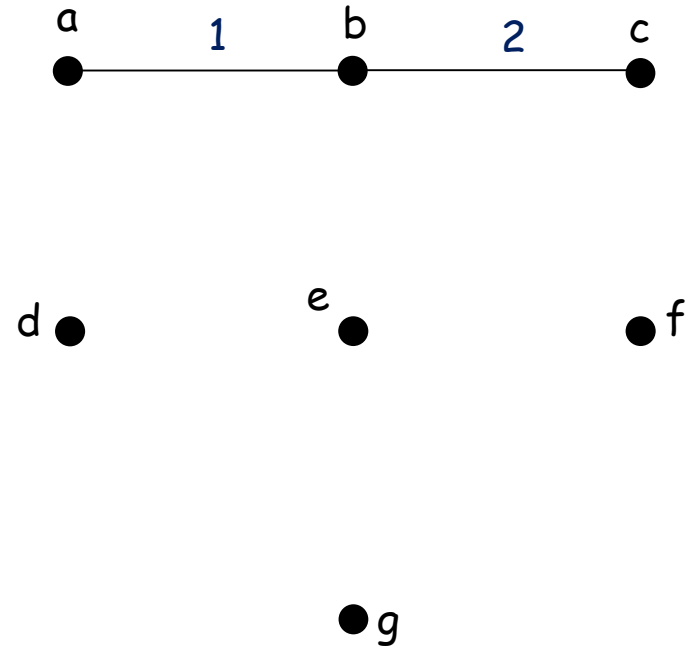
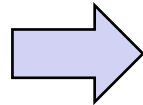
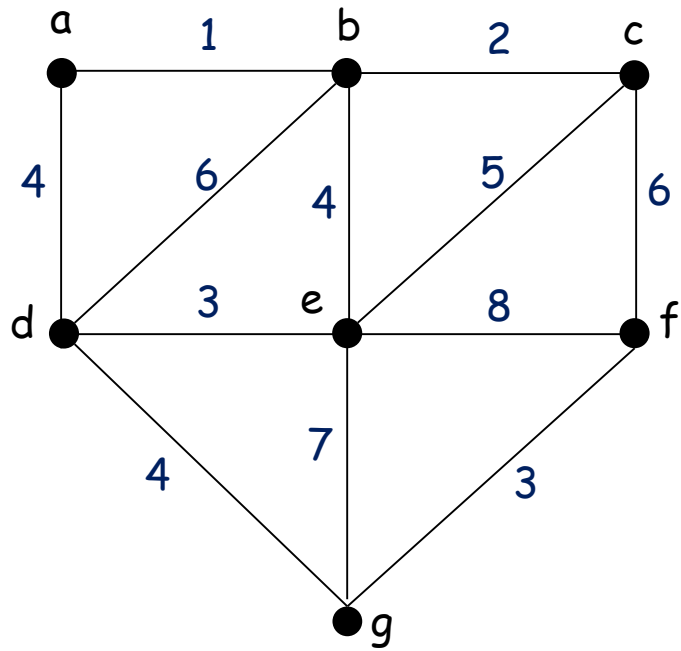
# Árboles



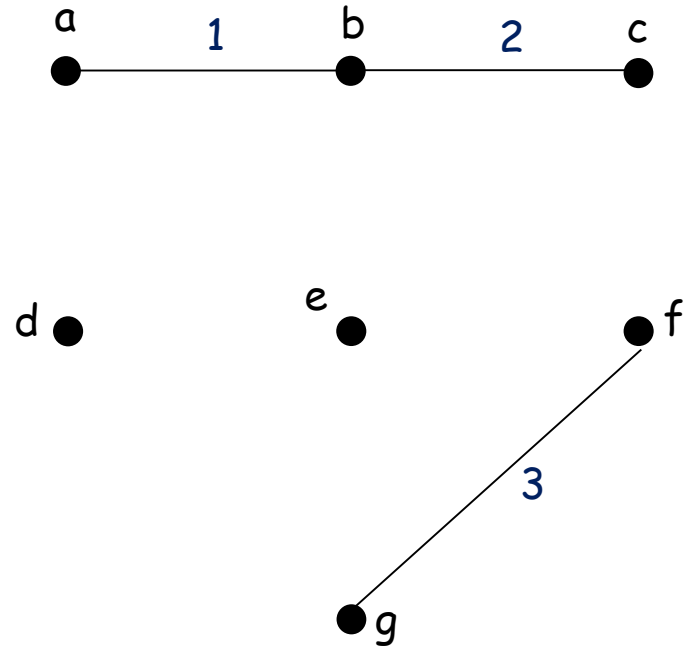
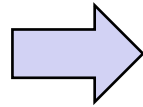
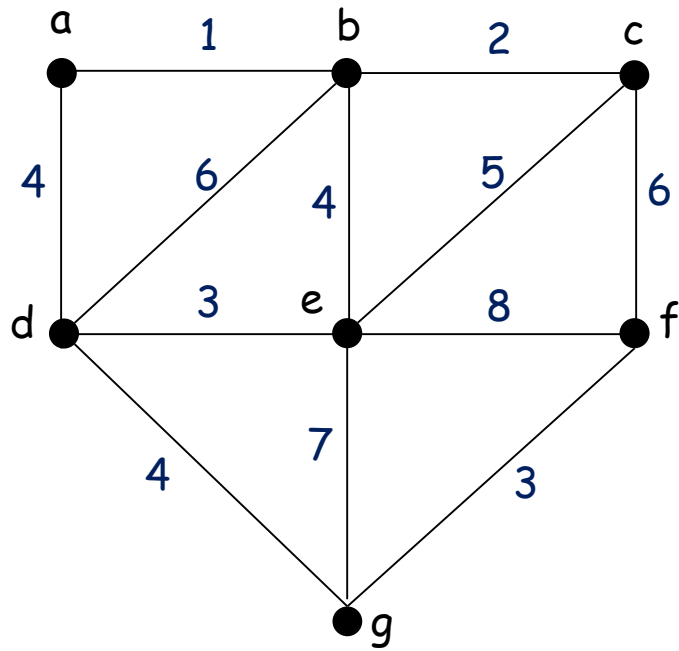
# Árboles



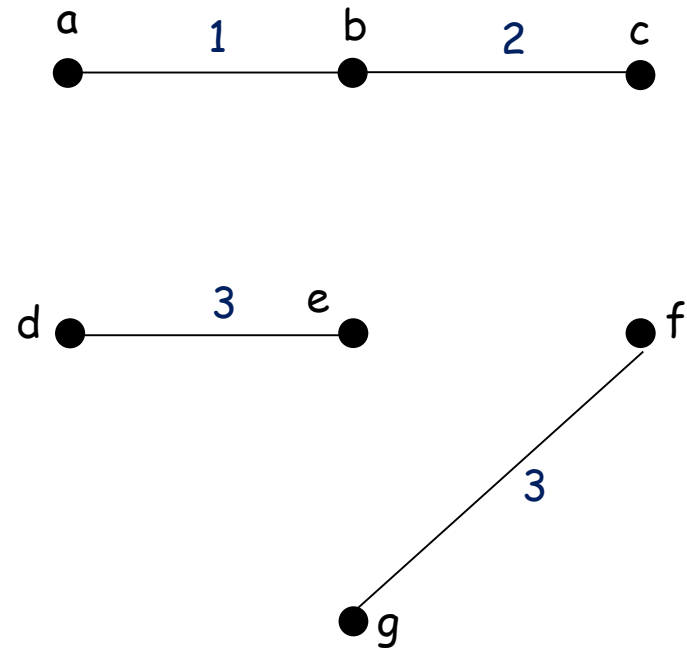
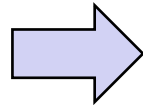
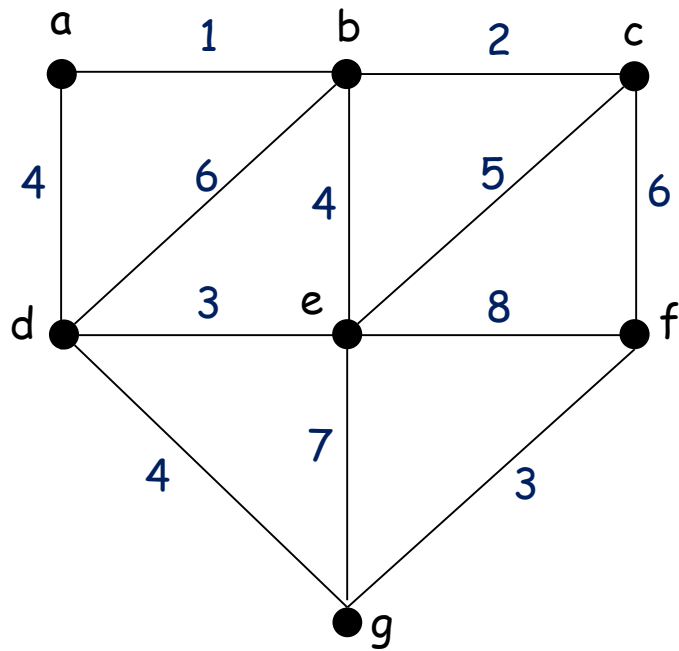
# Árboles



# Árboles

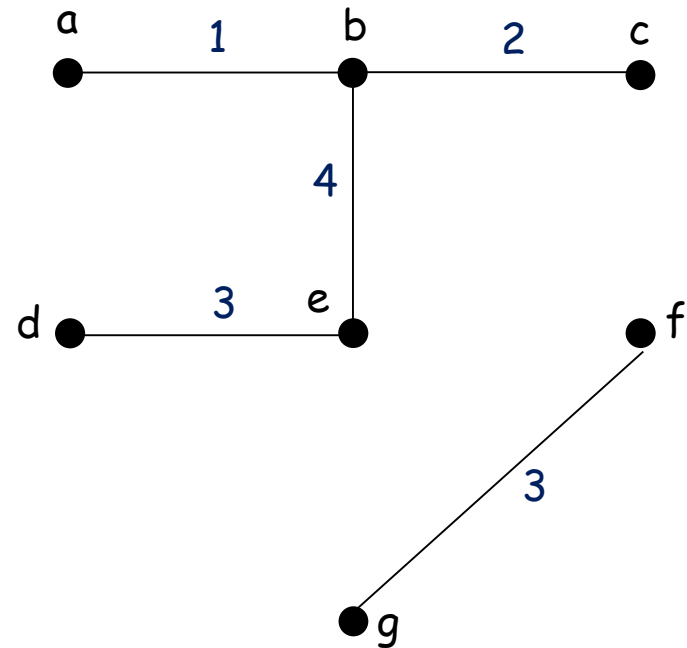
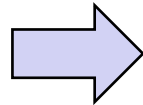
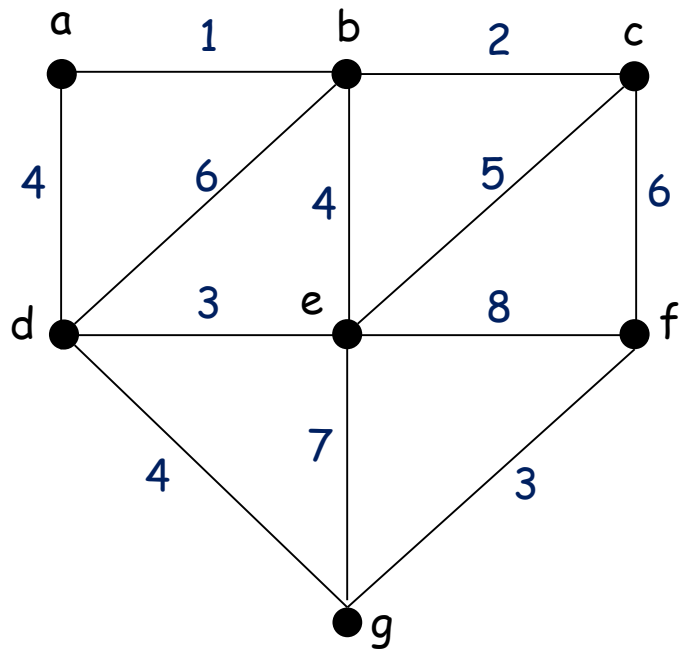


# Árboles

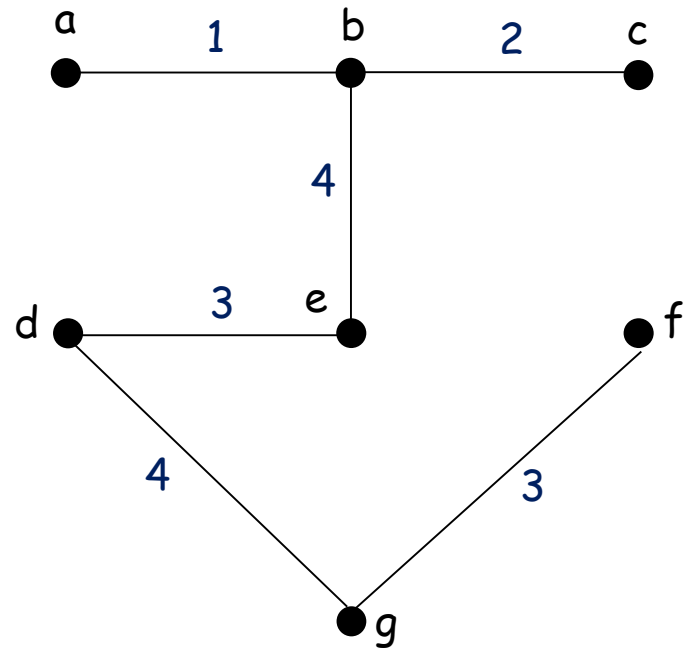
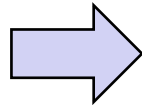
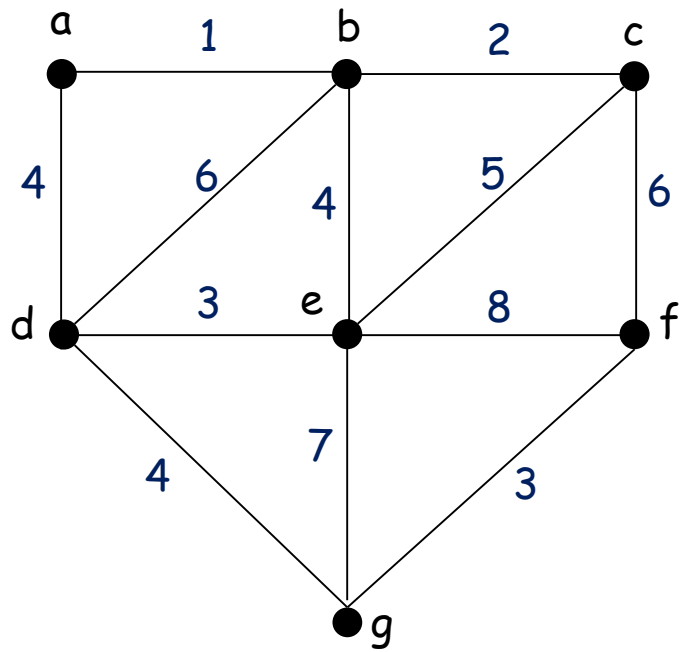




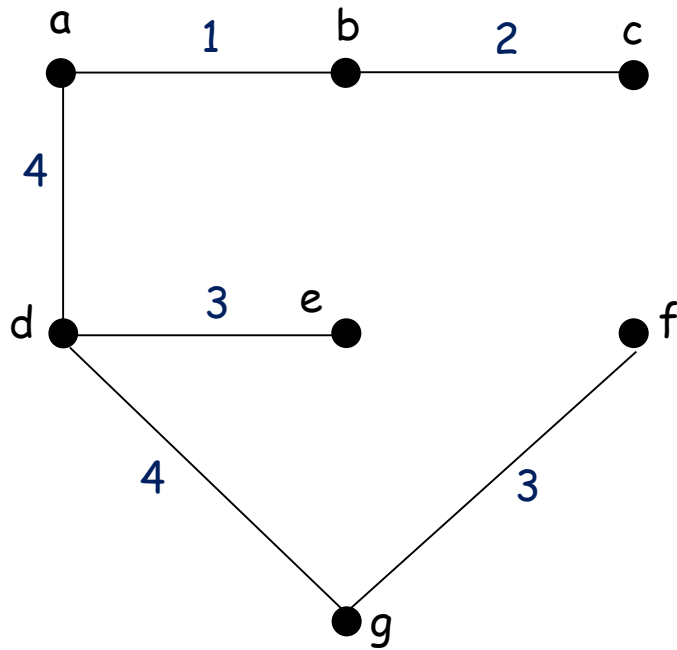
# Árboles



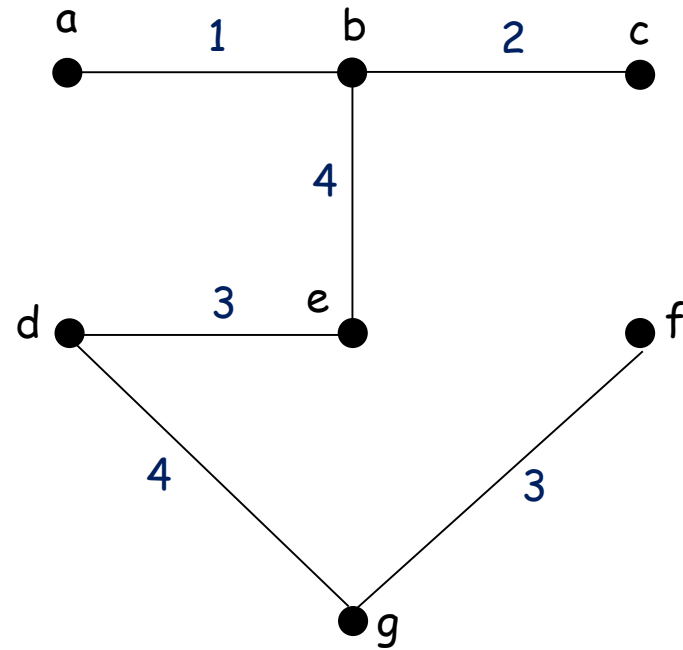
# Árboles



# Árboles



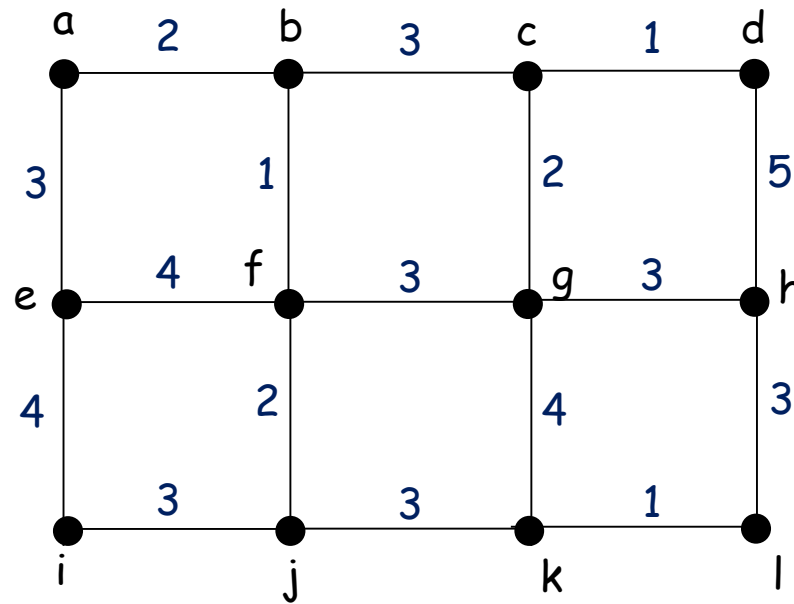
Árbol recubridor  
mínimo obtenido con el  
algoritmo de Prim



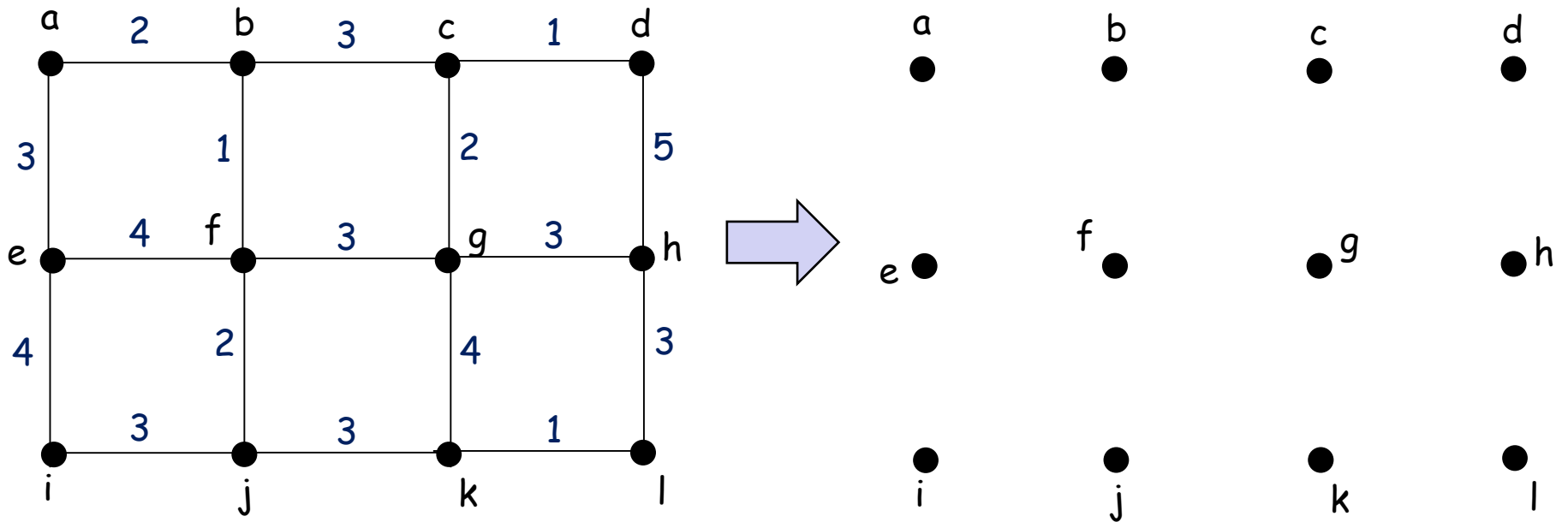
Árbol recubridor  
mínimo obtenido con el  
algoritmo de Kruskal

# Árboles

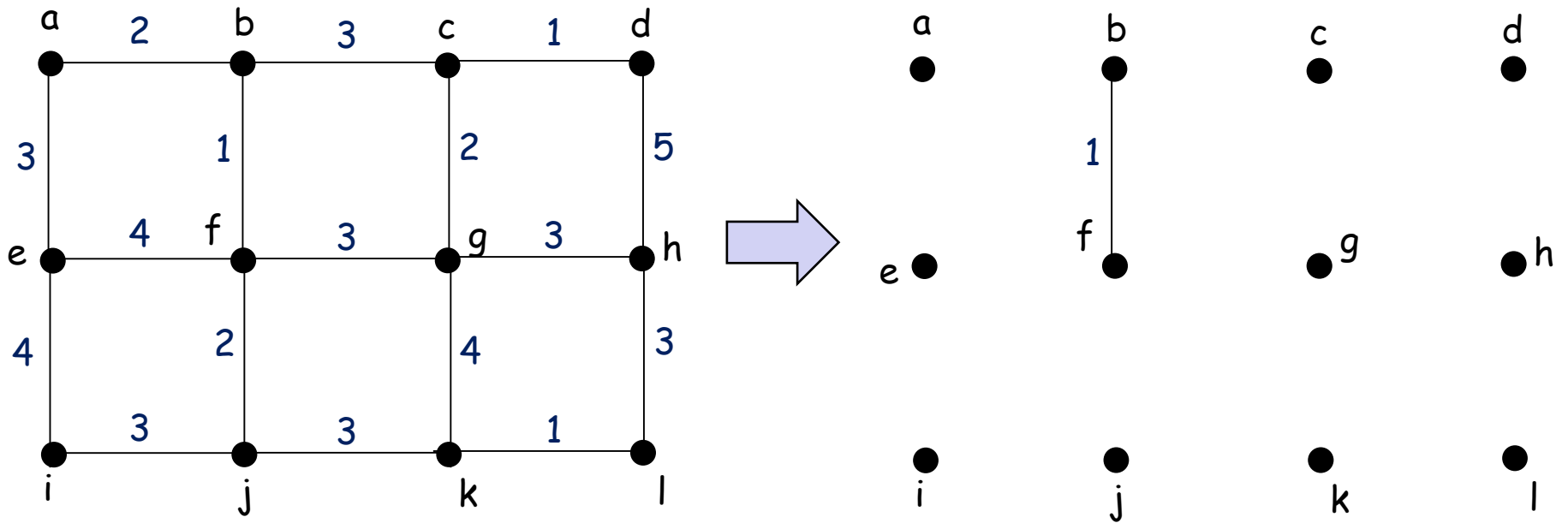
Encontrar un árbol recubridor mínimo usando el algoritmo de Kruskal



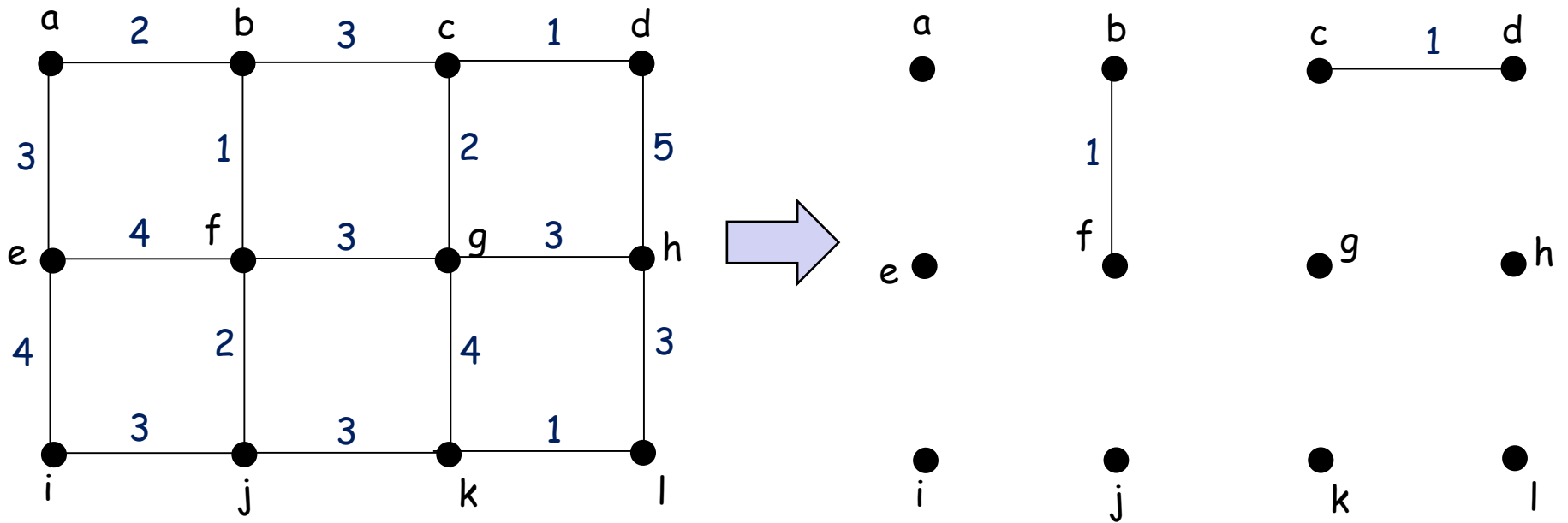
# Árboles



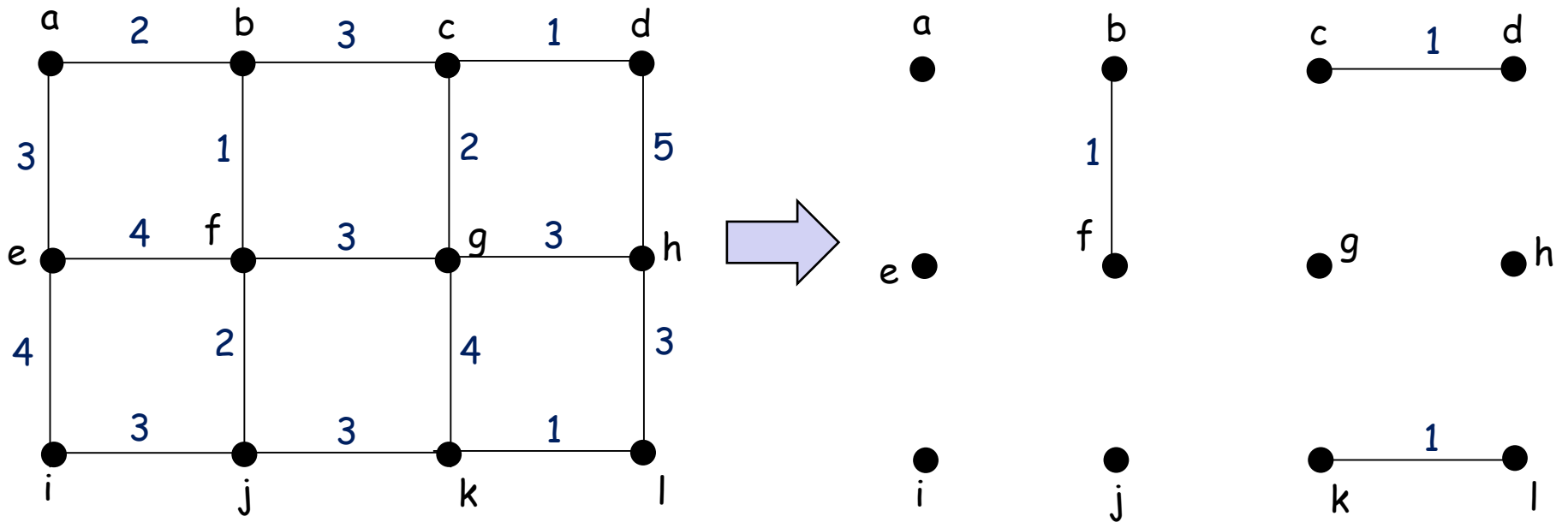
# Árboles



# Árboles

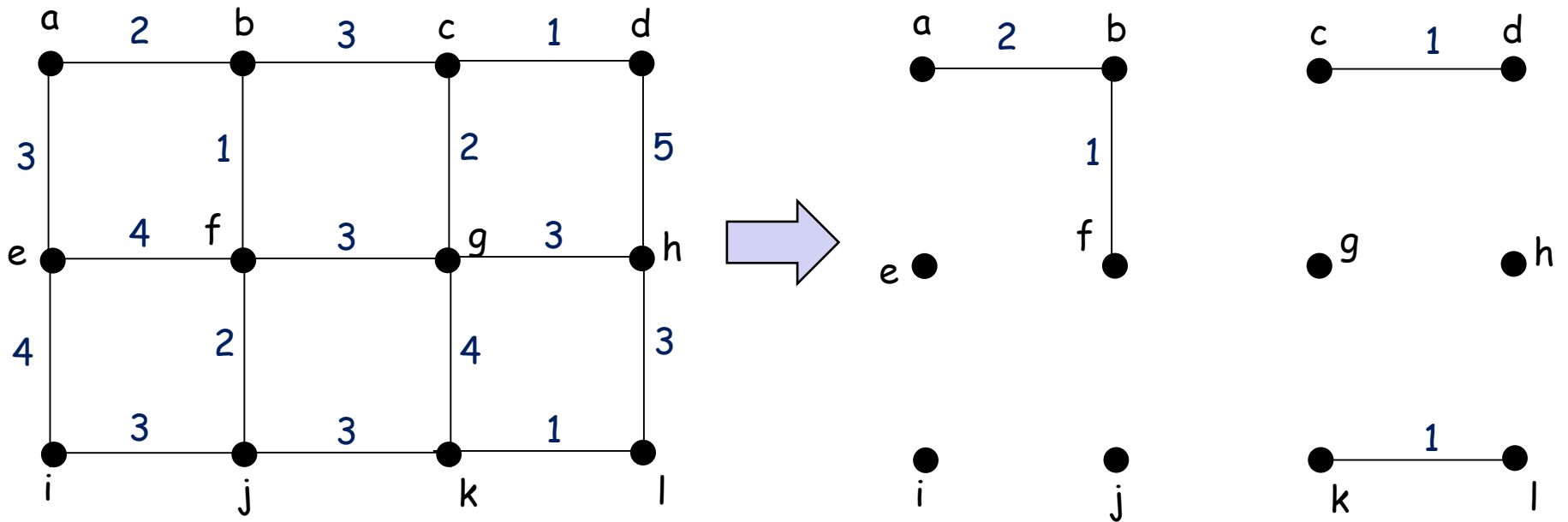


# Árboles

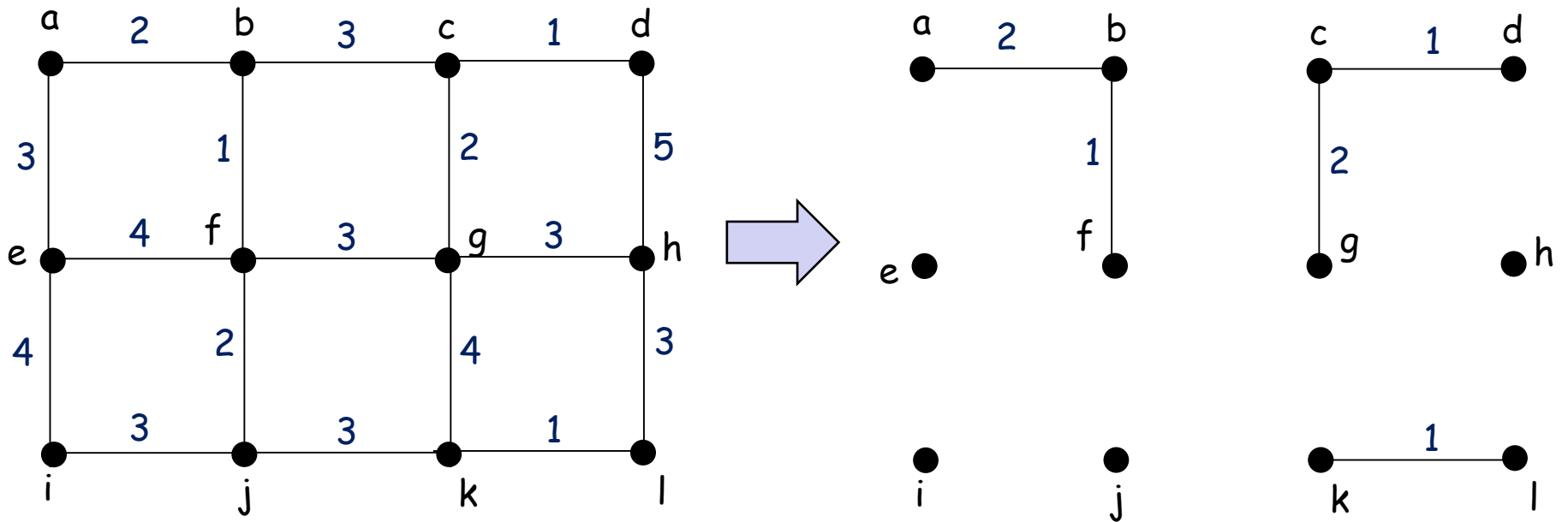




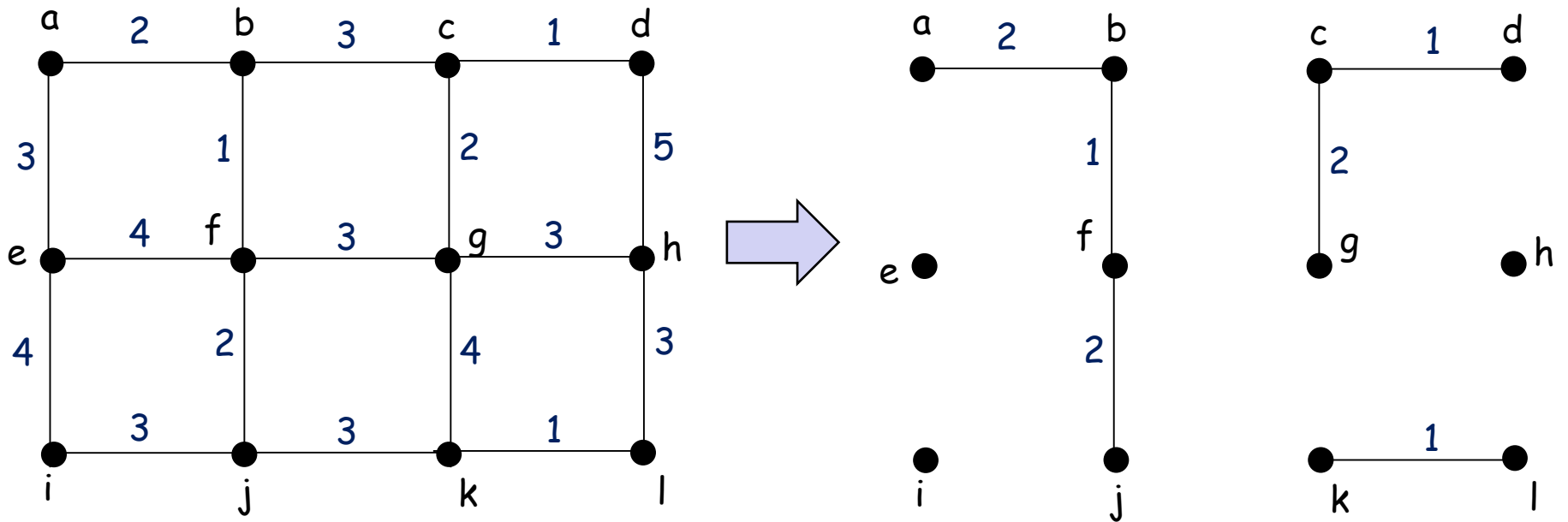
# Árboles



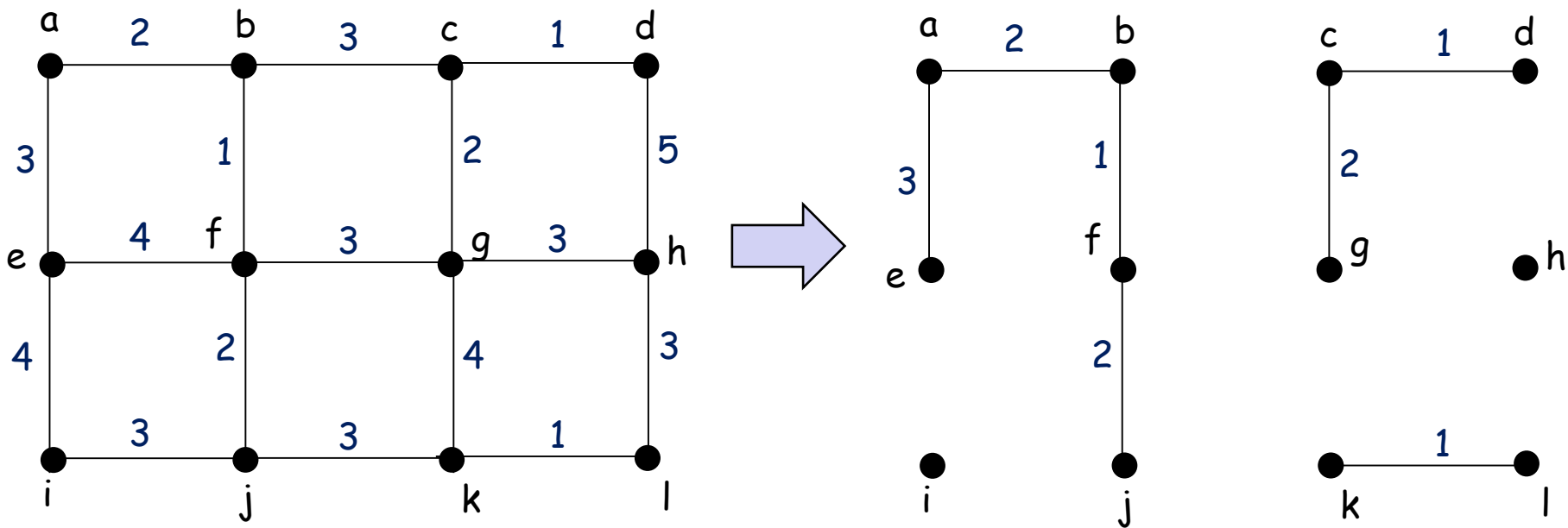
# Árboles



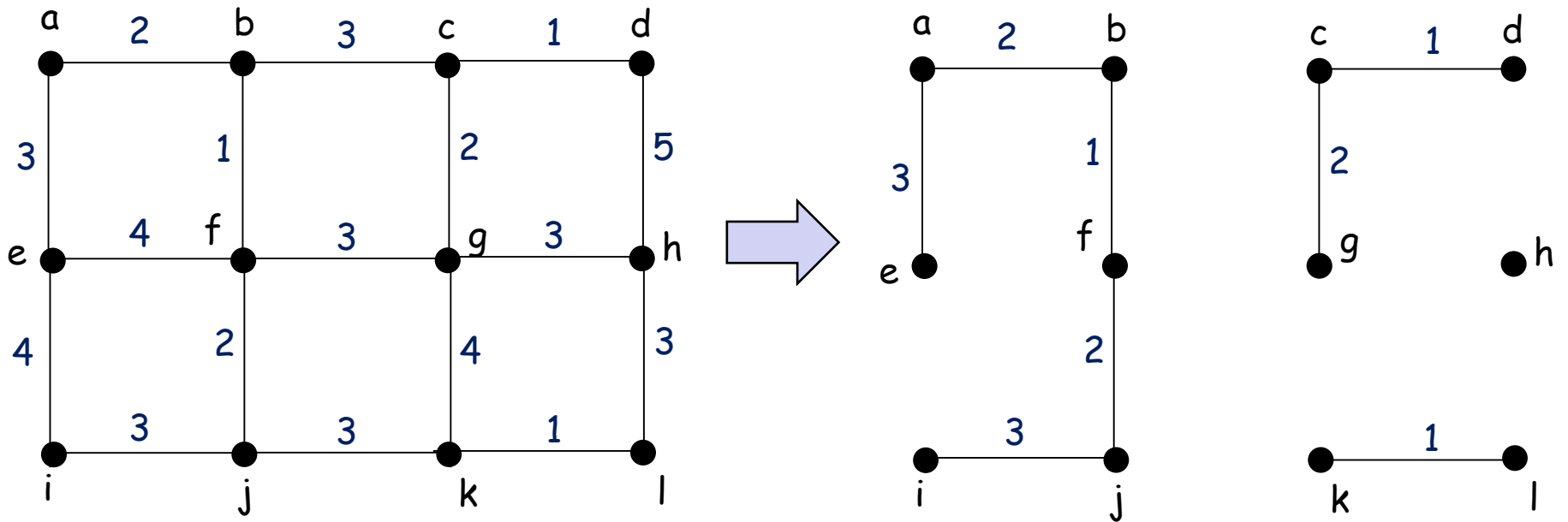
# Árboles



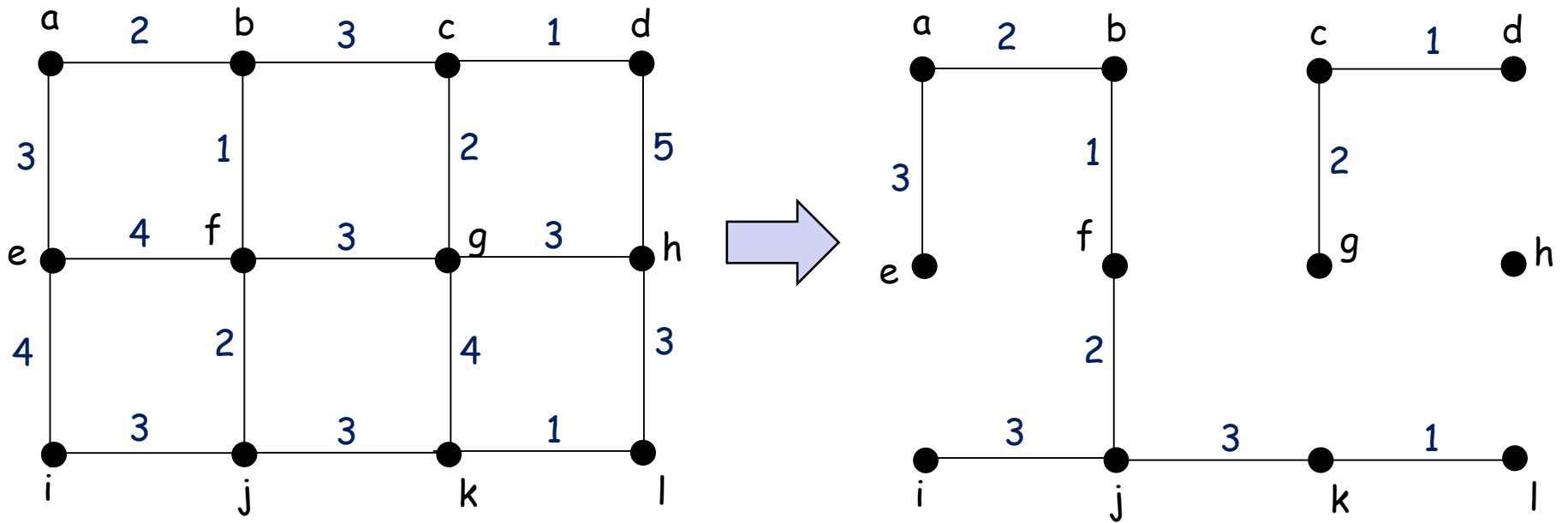
# Árboles



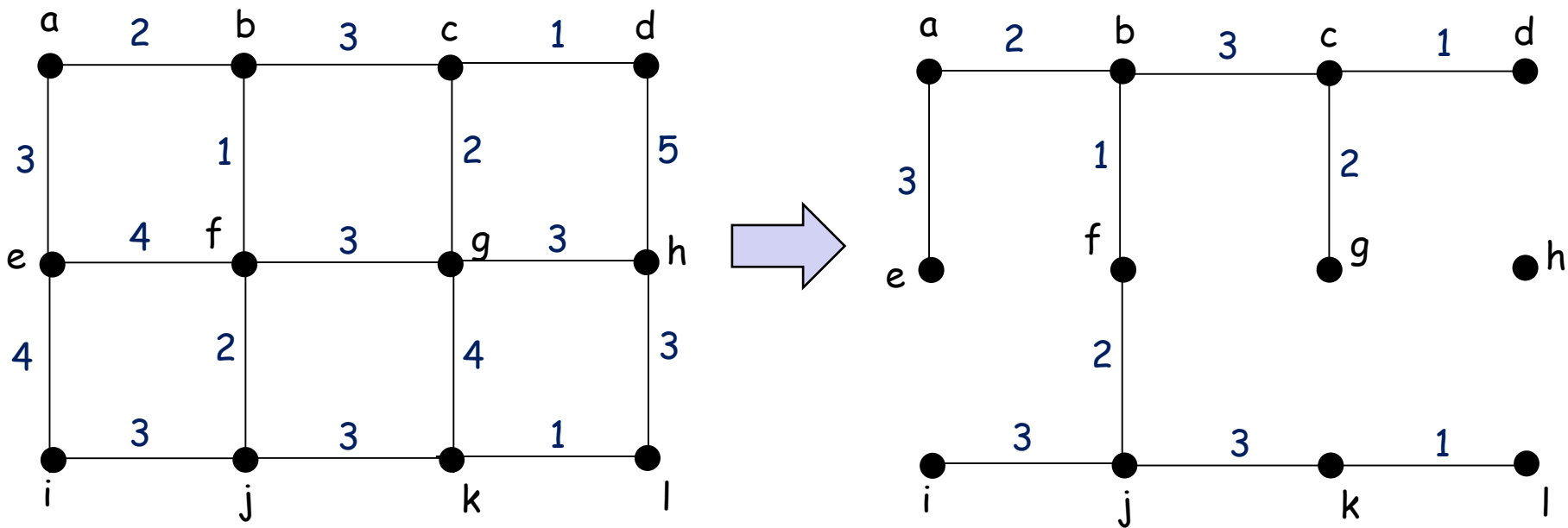
# Árboles



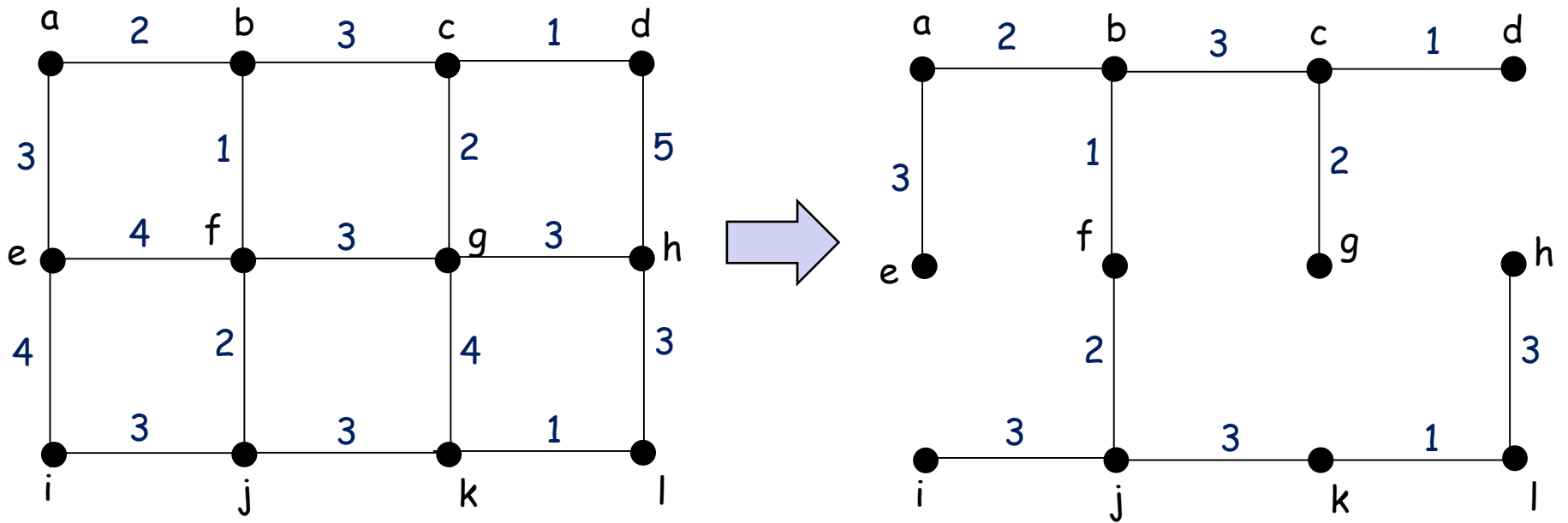
# Árboles



# Árboles

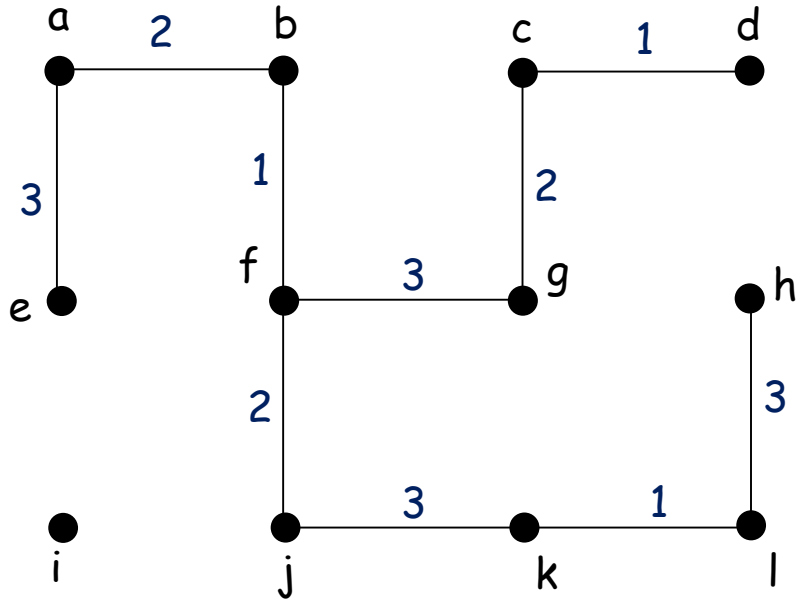


# Árboles

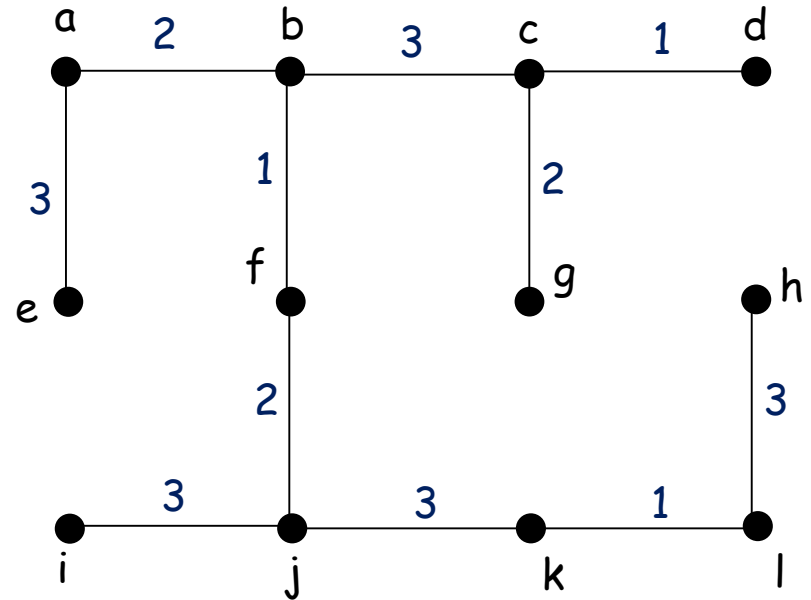




# Árboles



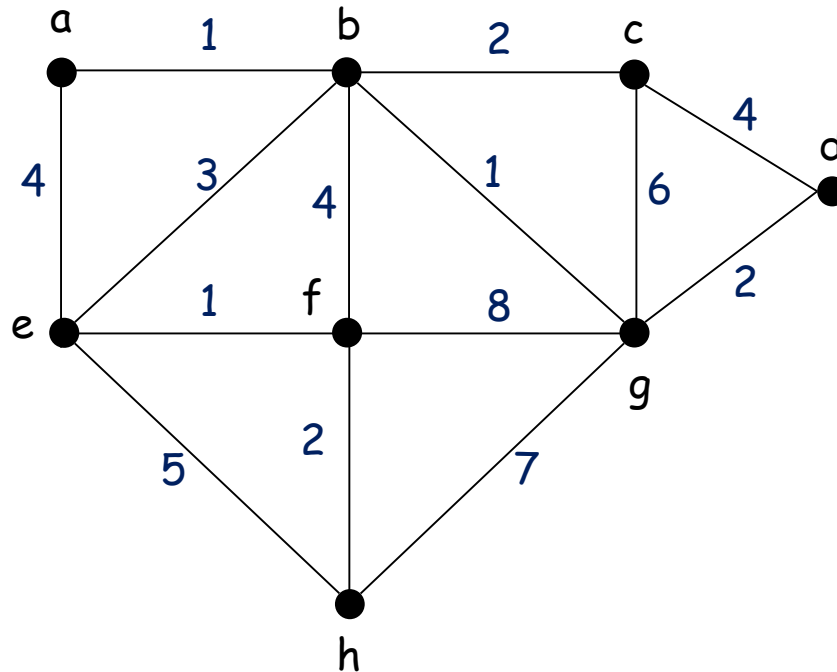
Árbol recubridor  
mínimo obtenido con el  
algoritmo de Prim



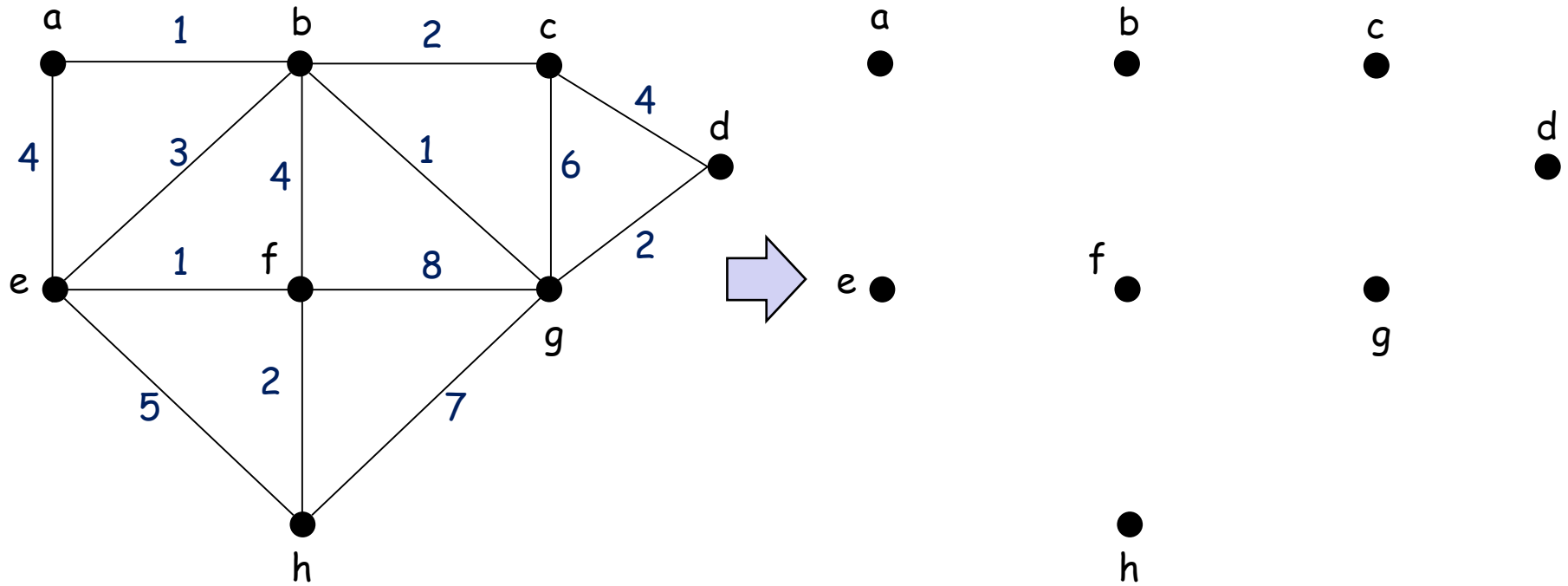
Árbol recubridor  
mínimo obtenido con el  
algoritmo de Kruskal

# Árboles

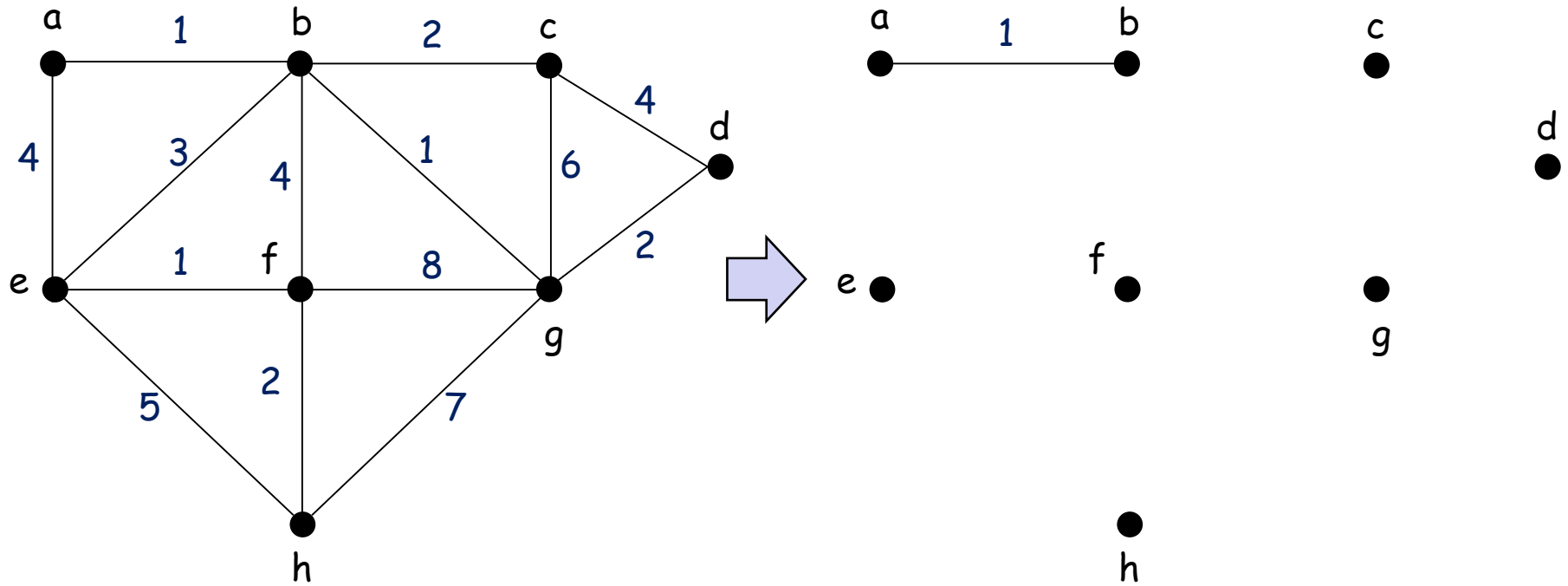
Encontrar un árbol recubridor mínimo usando el algoritmo de Kruskal



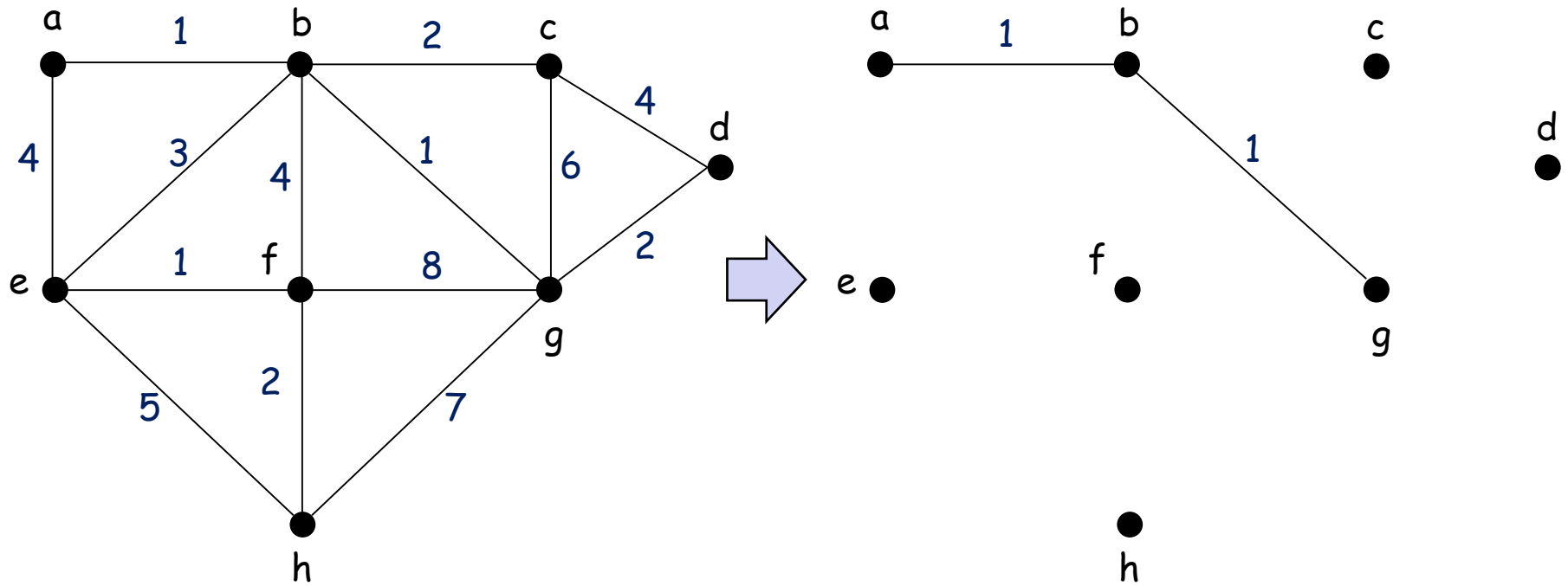
# Árboles



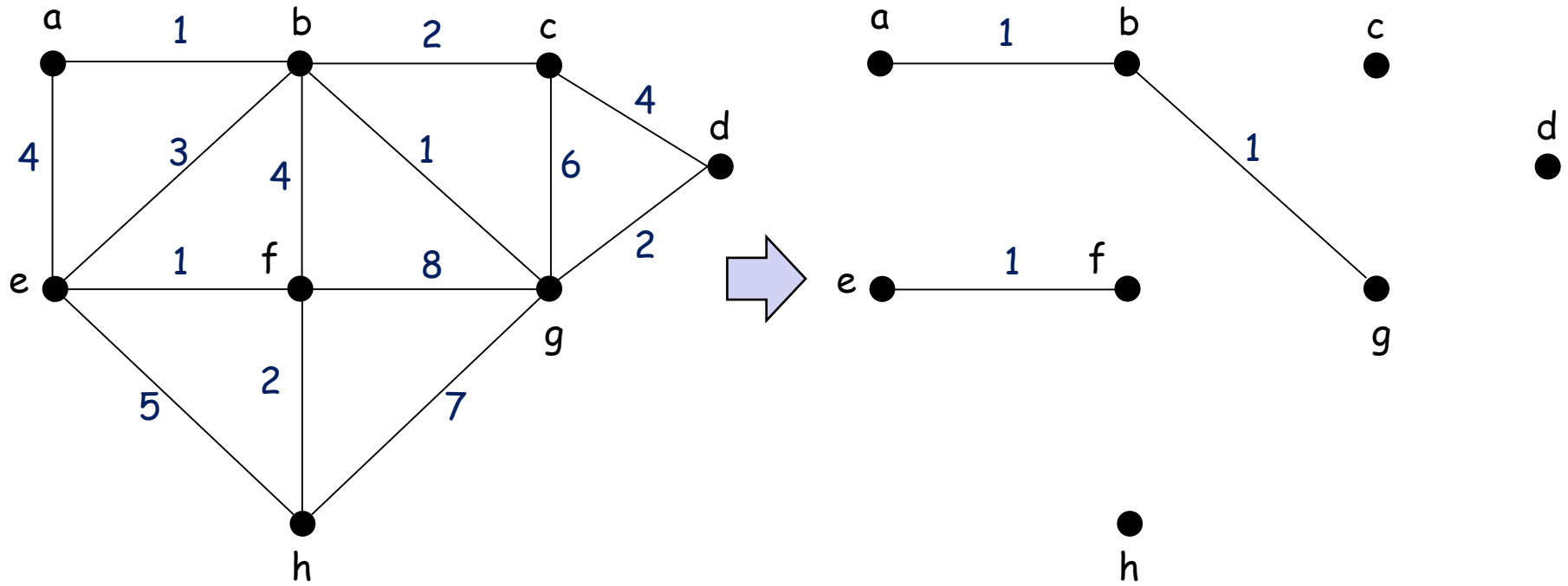
# Árboles



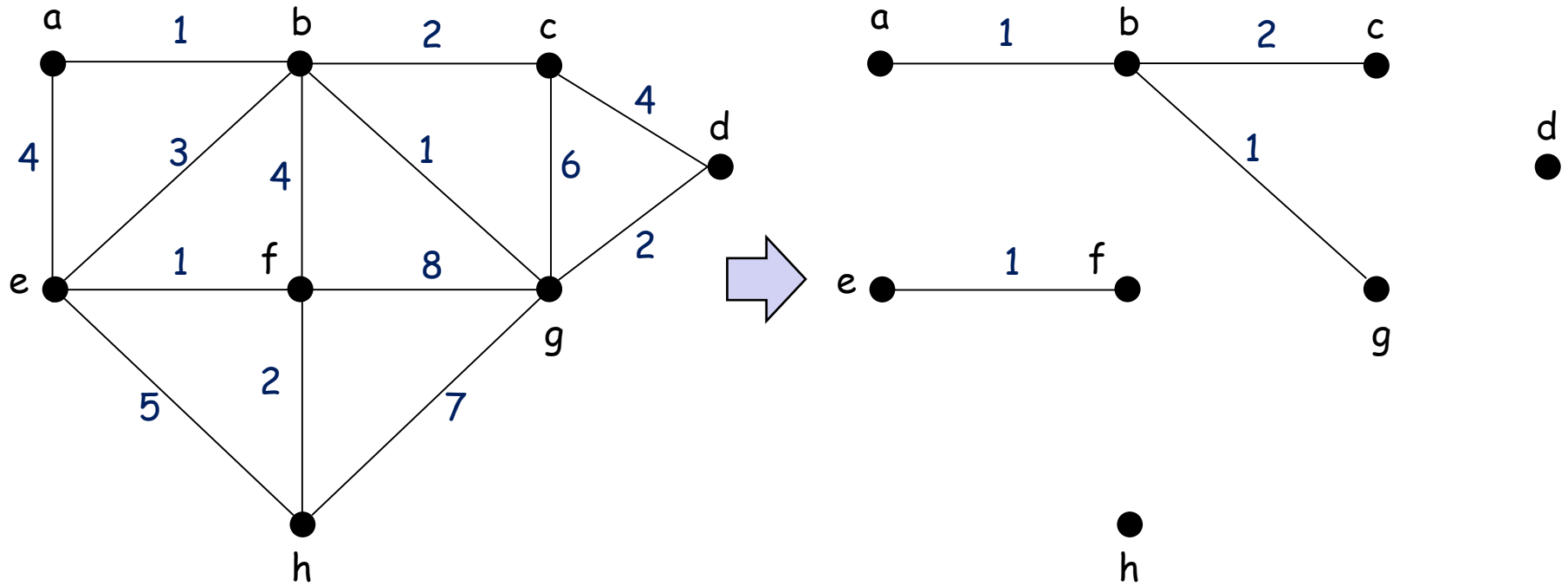
# Árboles



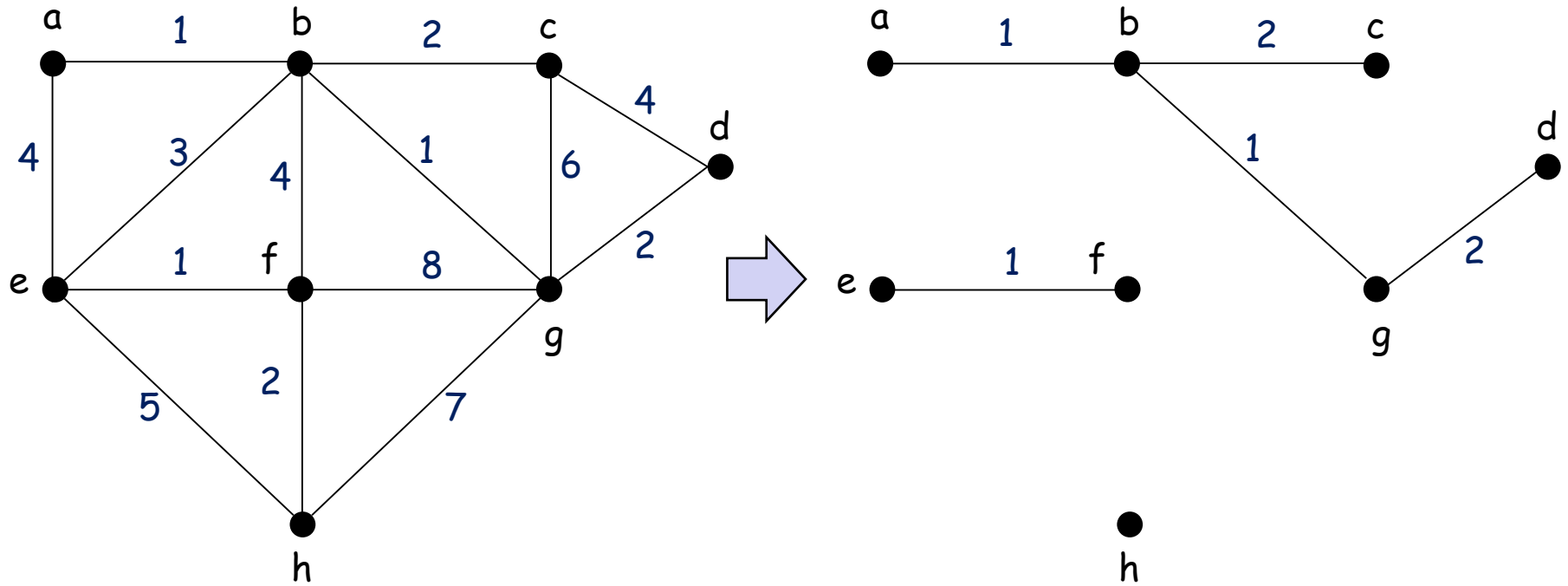
# Árboles



# Árboles

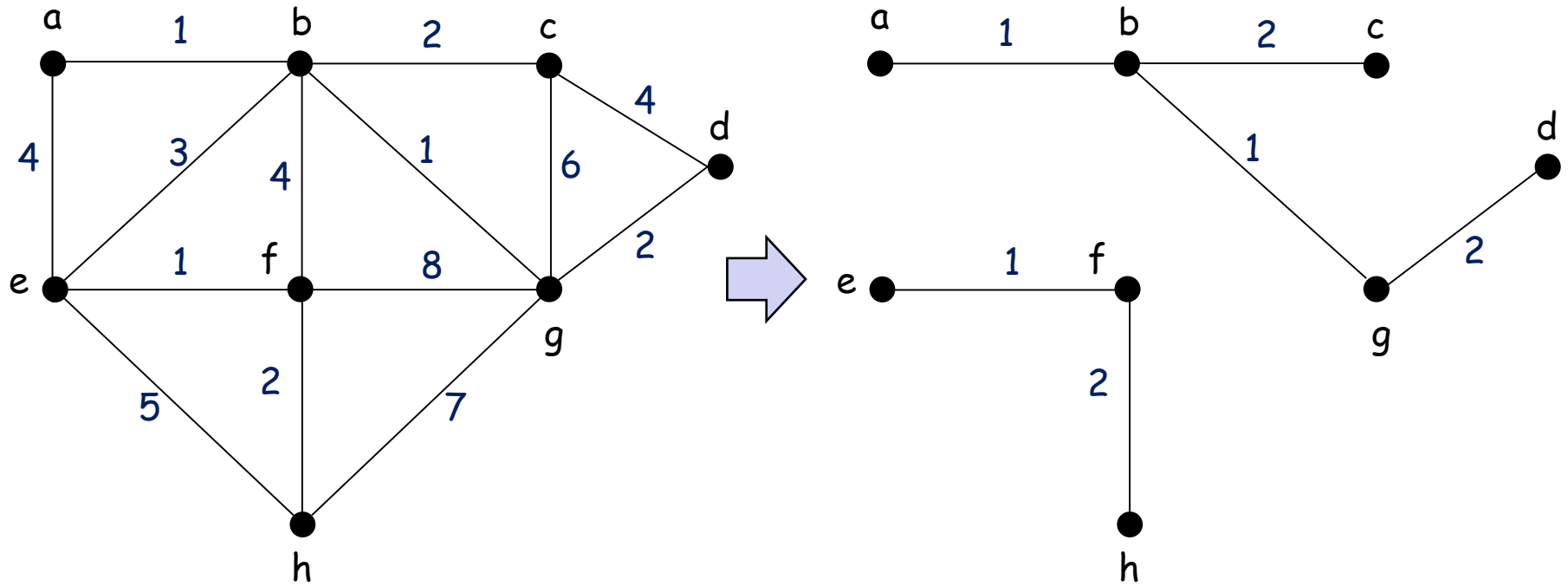


# Árboles

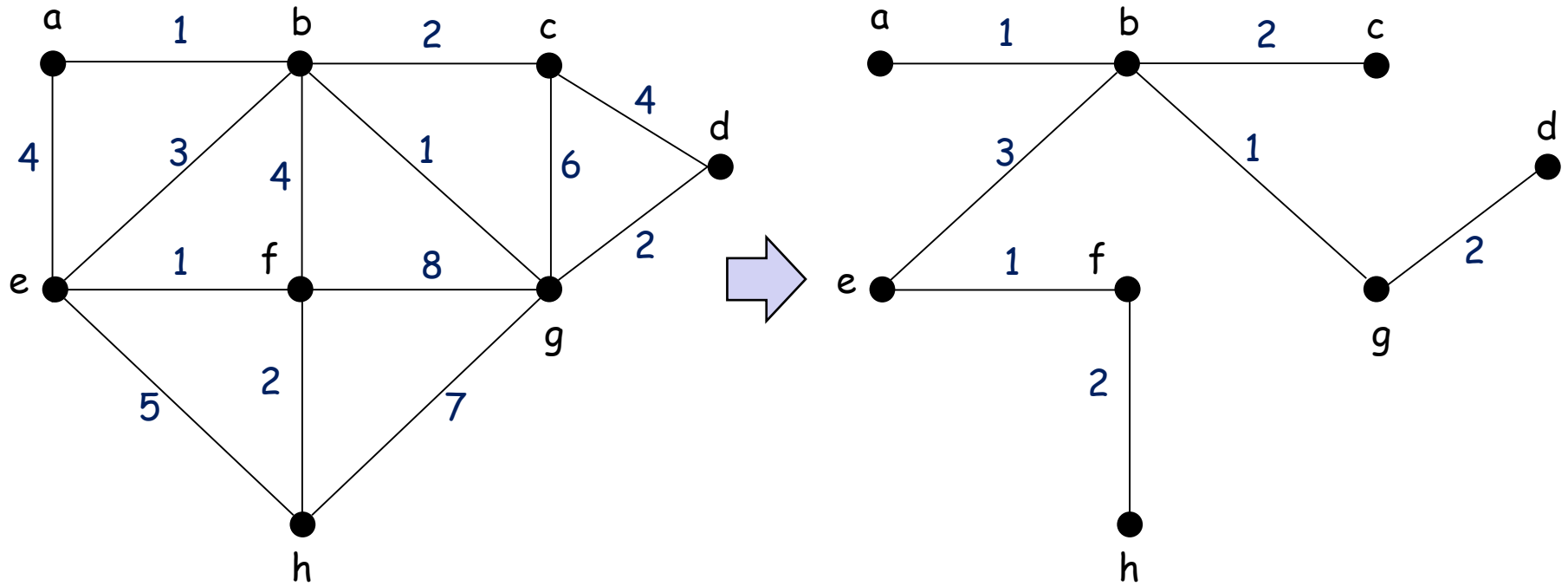




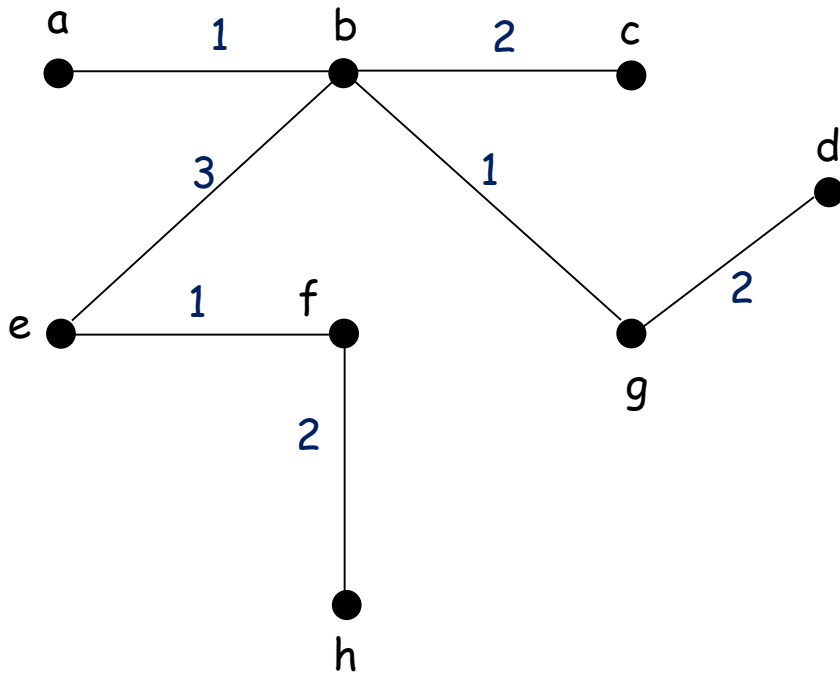
# Árboles



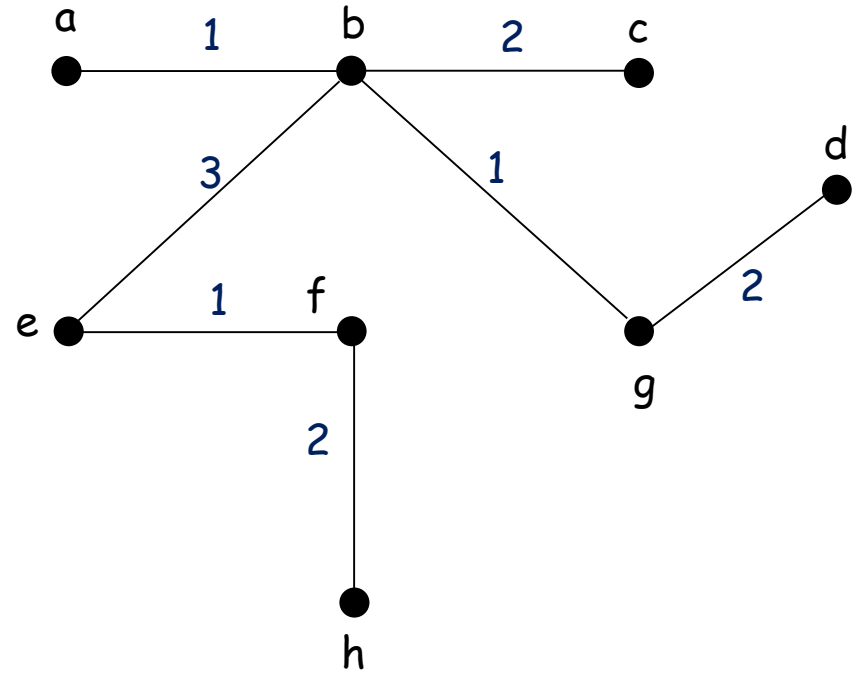
# Árboles



# Árboles



Árbol recubridor  
mínimo obtenido con el  
algoritmo de Prim



Árbol recubridor  
mínimo obtenido con el  
algoritmo de Kruskal