



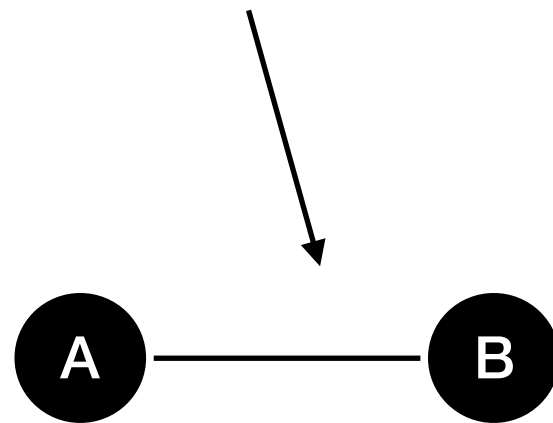
# Grafs I

Algorísmica Avançada | Enginyeria Informàtica

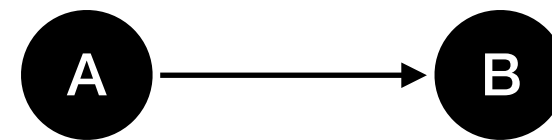
Santi Seguí | 2019-2020

# Què és un graf?

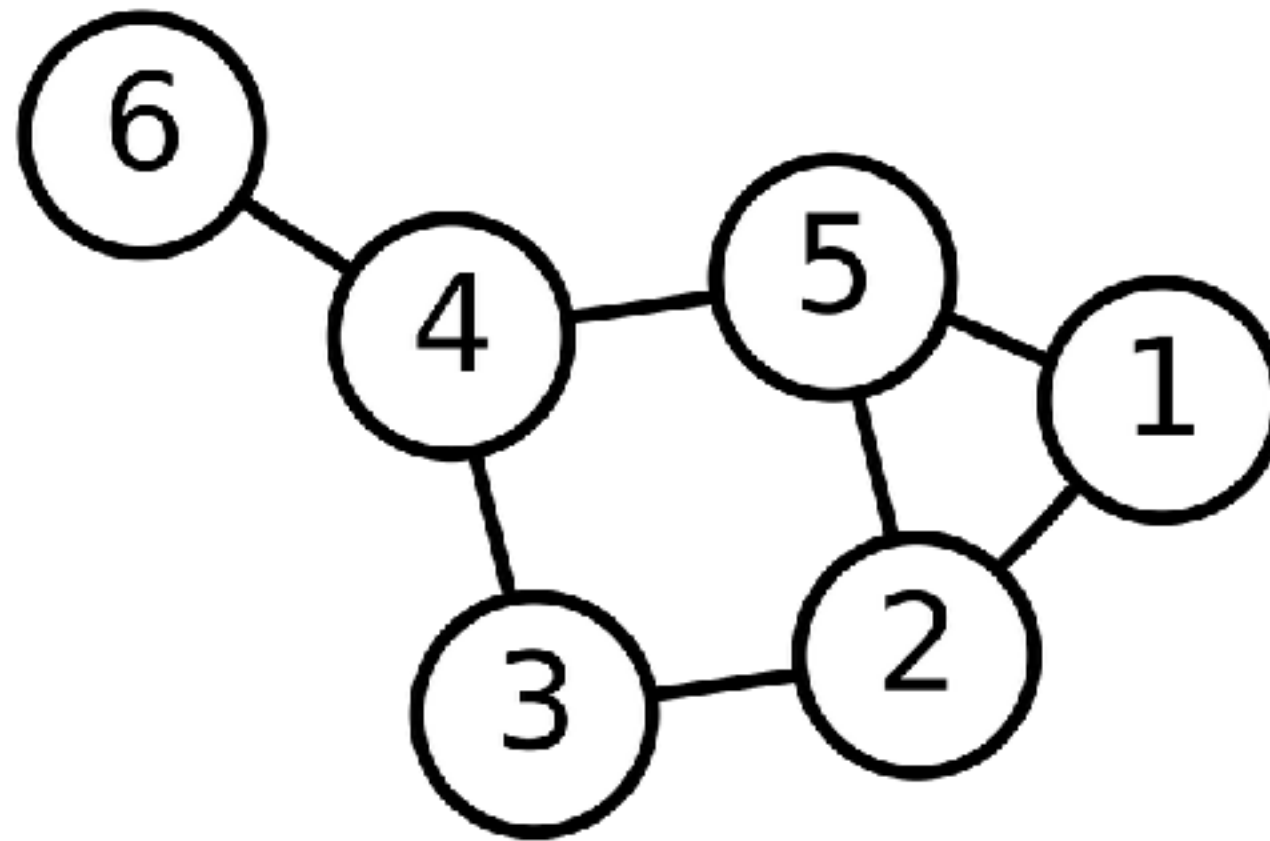
- Un **graf** és un conjunt de vèrtex o punts i un conjunt de d'arestes o arcs relacionats entre aquests vertexs. Els grafs tenen una estructura topològica que ens permet establir relacions espacials entre els vèrtexs que estem estudiant.
- Un **graf** és un conjunt ordeants  $G = (V, E)$  on:
  - $V$  és el conjunt de vertex
  - $E \subseteq \{\{x, y\} \mid (x, y) \in V^2 \wedge x \neq y\}$  és un conjunt d'arestes



**Graf no dirigit**



**graf dirigit**



$$V := \{1, 2, 3, 4, 5, 6\}$$

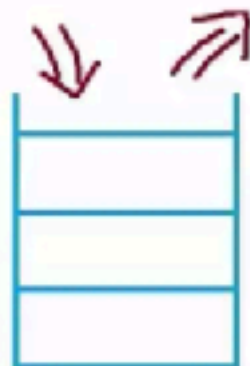
$$E := \{\{1, 2\}, \{1, 5\}, \{2, 3\}, \{2, 5\}, \{3, 4\}, \{4, 5\}, \{4, 6\}\}.$$

# Què és un graf?

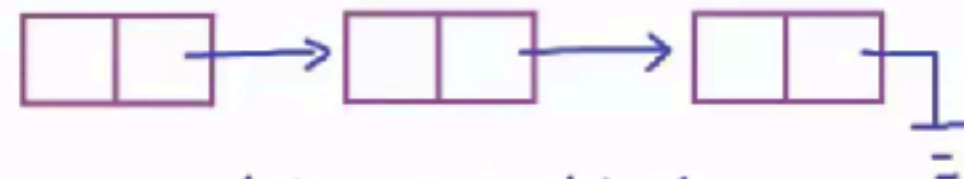
- El graf és una estructura de dades **NO LINEAL**

## Estructures de dades Lineals

Array



Stack



Linked List

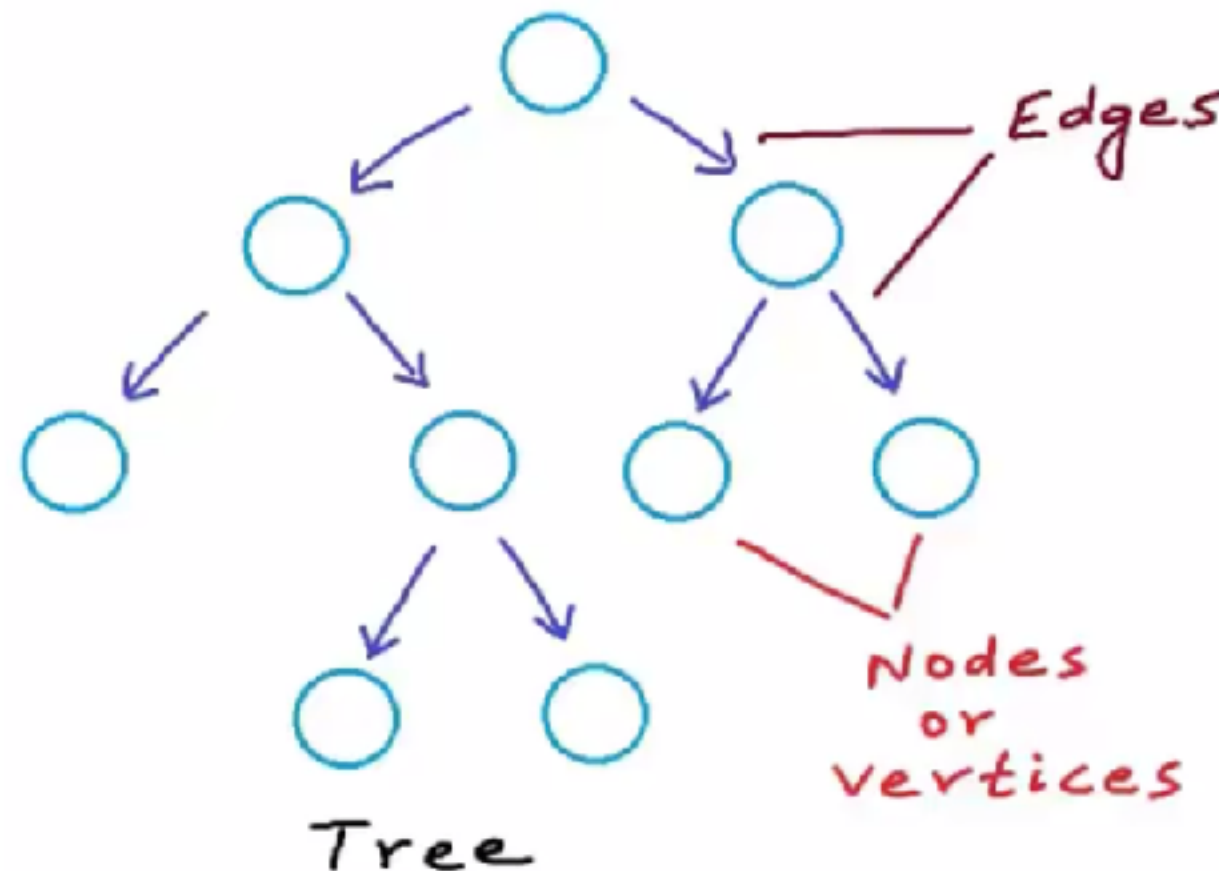


Queue

# Què és un graf?

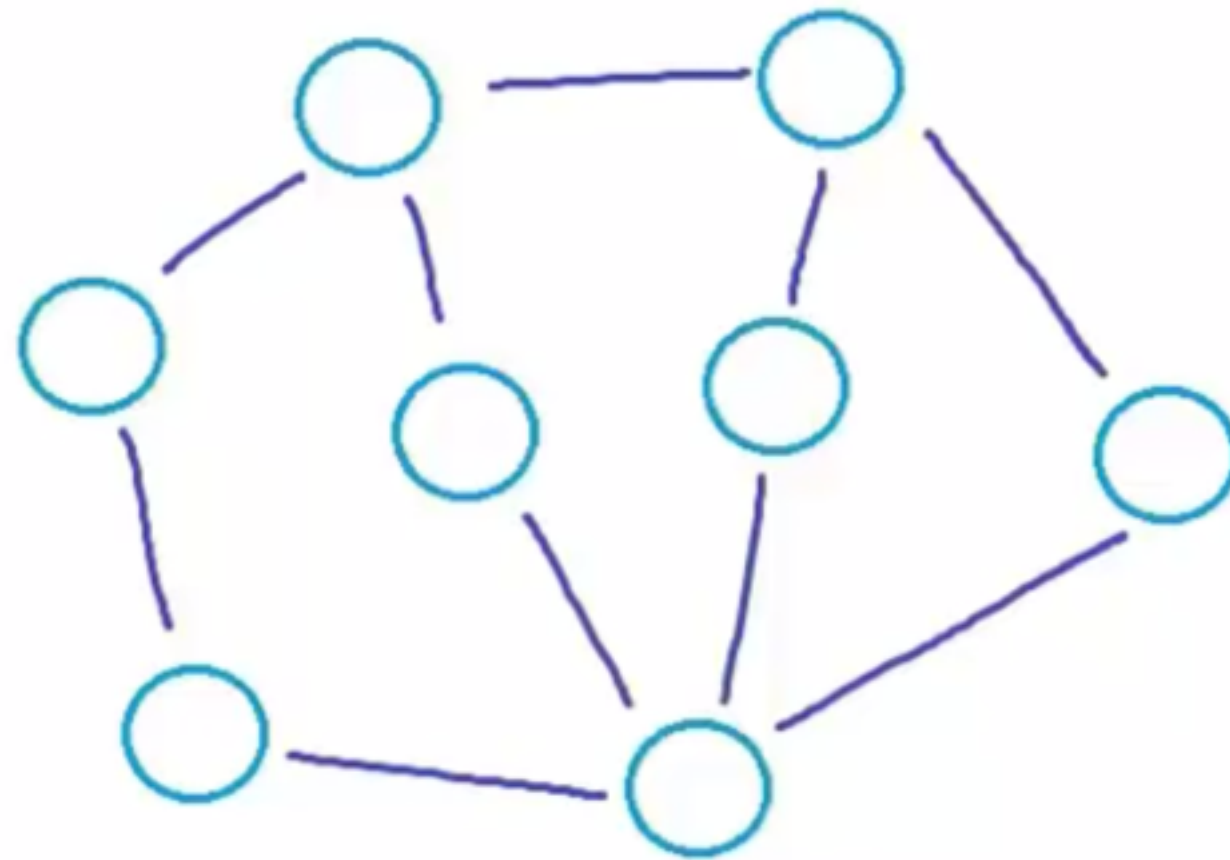
- El graf és una estructura de dades **NO LINEAL**

Quina estructura de dades **NO LINEAL** coneixeu?



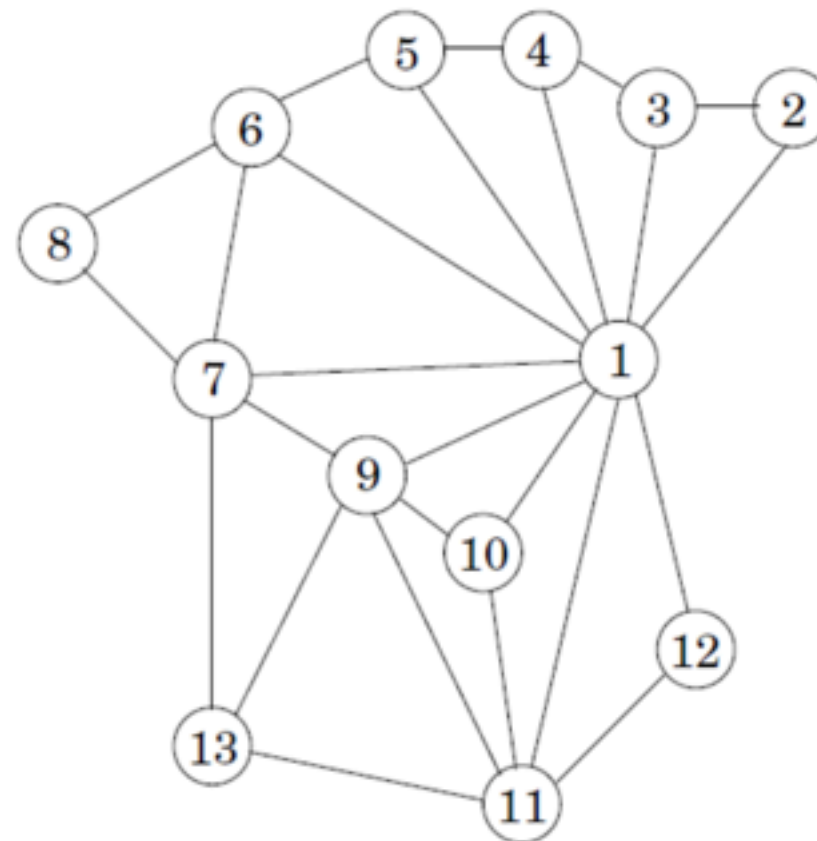
# Què és un graf?

- El graf és una estructura de dades **NO LINEAL**



# Algoritmes sobre GRAFS

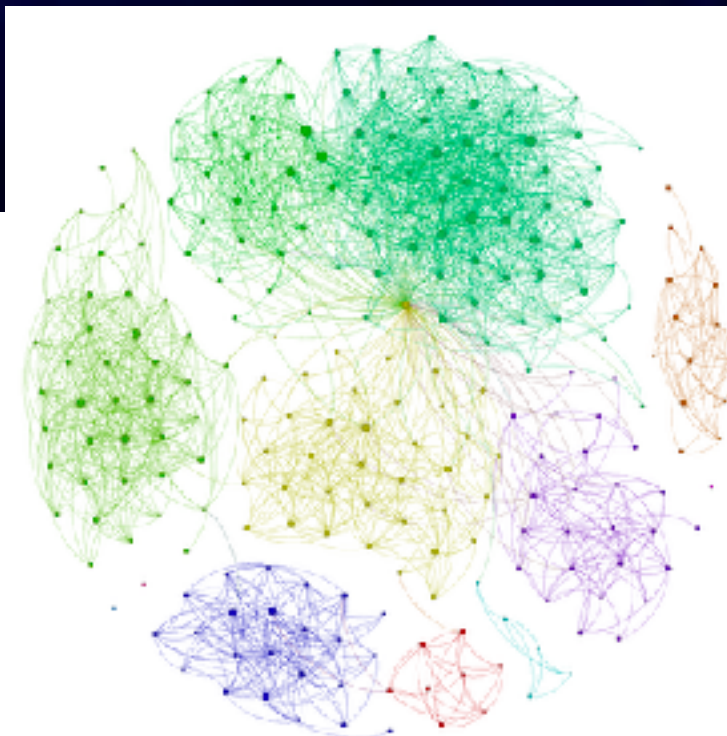
- Per a què serveix un gràf?

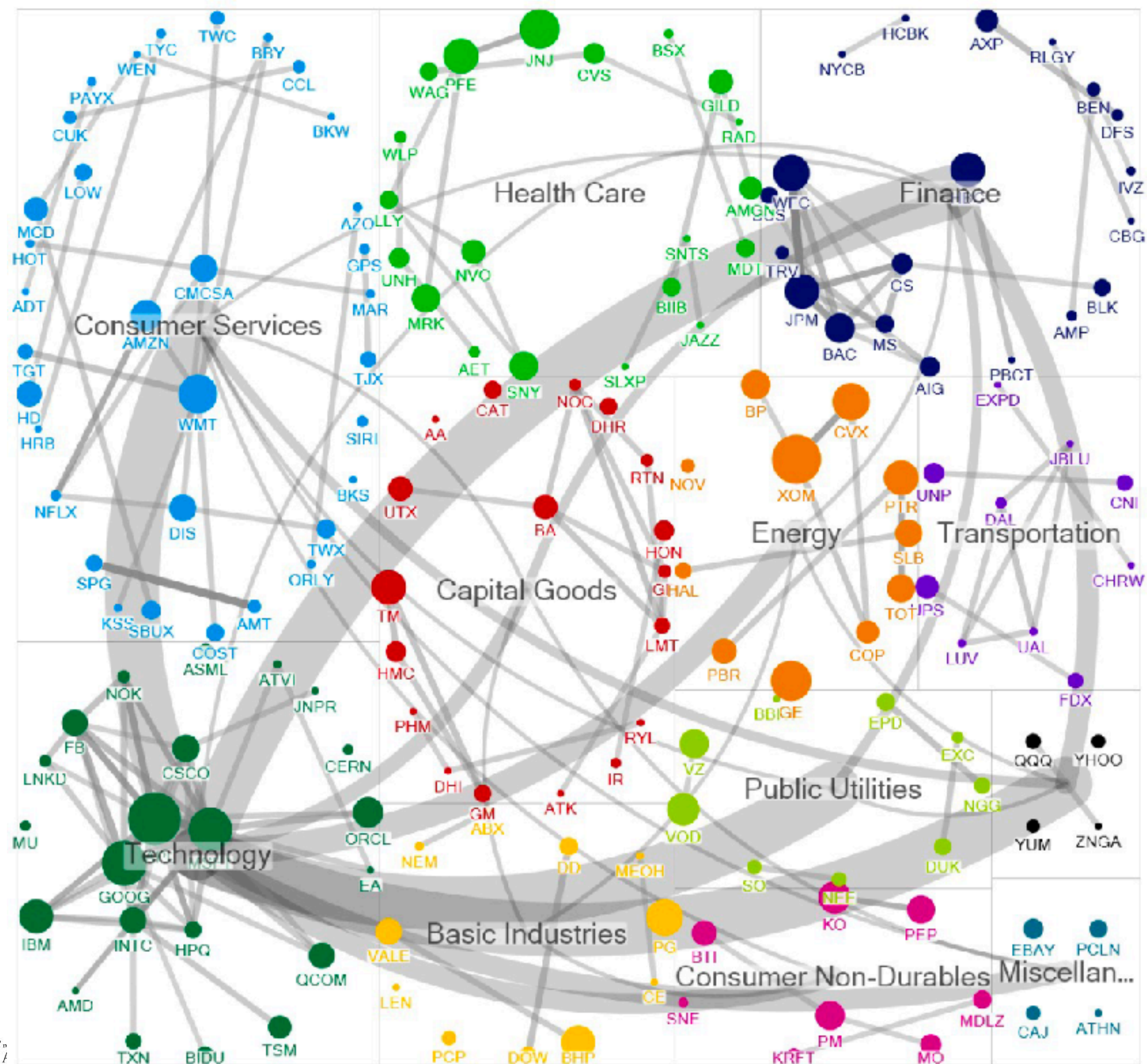














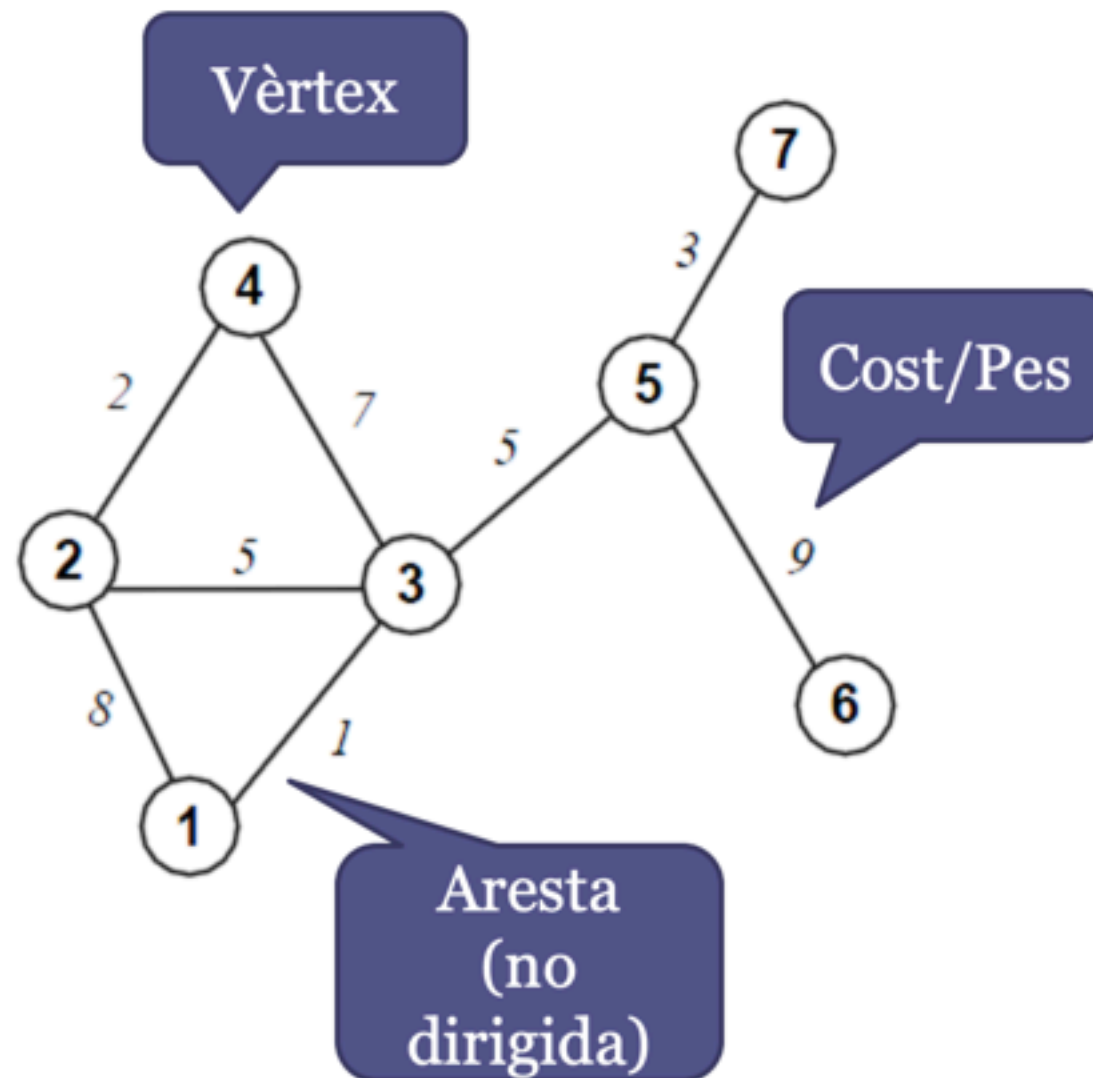
# Aplicacions amb grafes

Graph	Vertex	Edge
Communication	telephone, computer	fiber optic cable
circuit	gate, register, processor	wire
financial	stock, currency	transactions
transportation	street intersection, airport	highway, airway route
game	board position	legal move
social relationship	person, actor	friendship, movie cast
molecule	atom	bond
neural network	neuron	synapse

# Algoritmes sobre GRAFS

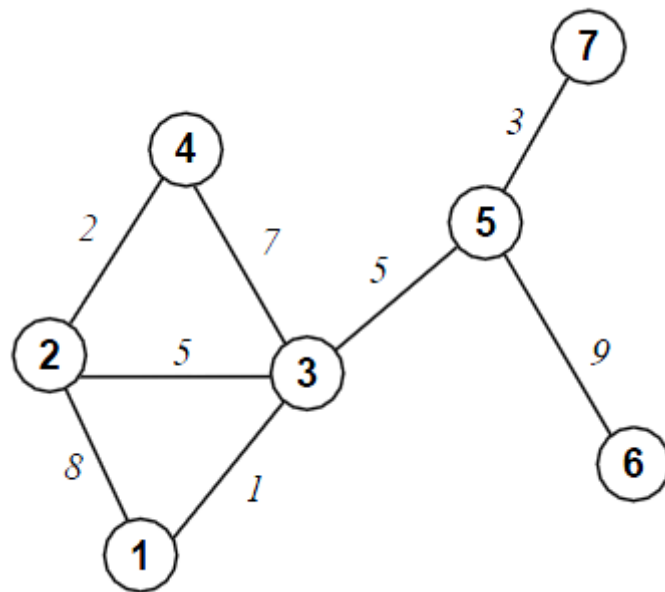
Com representem un graf?

$$G=(V,E)$$



# Algorismes sobre grafos

- Com representem un graf? **Estructura de graf**
- Matriu d'adjacència

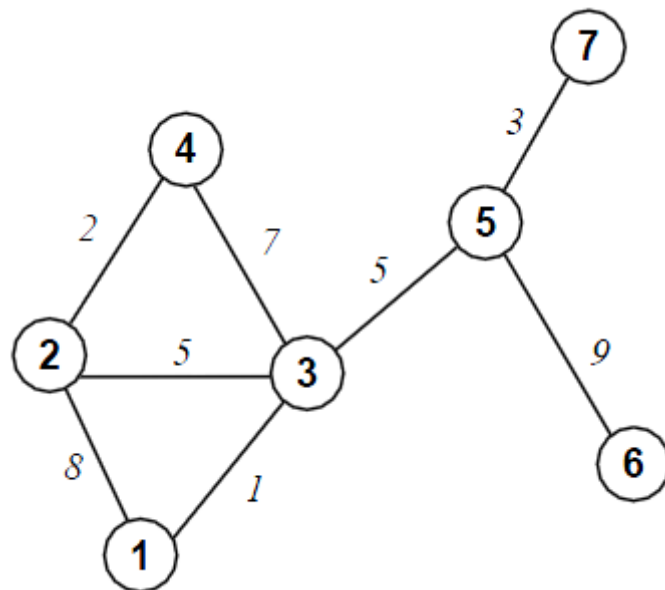


	1	2	3	4	5	6	7
1	0	8	1	$\infty$	$\infty$	$\infty$	$\infty$
2	8	0	5	2	$\infty$	$\infty$	$\infty$
3	1	5	0	7	5	$\infty$	$\infty$
4	$\infty$	2	7	0	$\infty$	$\infty$	$\infty$
5	$\infty$	$\infty$	5	$\infty$	0	9	3
6	$\infty$	$\infty$	$\infty$	$\infty$	9	0	$\infty$
7	$\infty$	$\infty$	$\infty$	$\infty$	3	$\infty$	0

Sense pesos  $\rightarrow a_{ij} = \begin{cases} 1 & \text{if there is an edge from } v_i \text{ to } v_j \\ 0 & \text{otherwise.} \end{cases}$

# Algorismes sobre grafos

- Com representem un graf? **Estructura de graf**
- Llista d'adjacència



1	●	→	(2,8)	(3,1)		
2	●	→	(1,8)	(3,5)	(4,2)	
3	●	→	(1,1)	(4,7)	(2,5)	(5,5)
4	●	→	(2,2)	(3,7)		
5	●	→	(3,5)	(7,3)	(6,9)	
6	●	→	(5,9)			
7	●	→	(5,3)			



# Algorismes sobre grafs

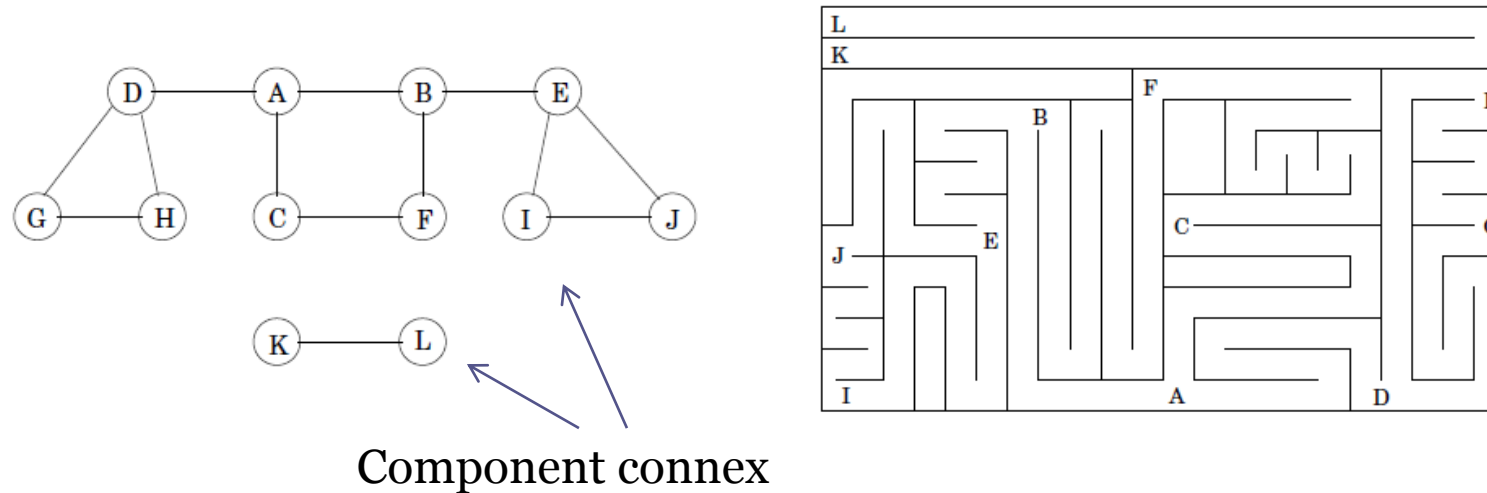
- Matriu ó llista d'adjacència?
  - Matriu  $\rightarrow |V|^2$  posicions  $\rightarrow$  un accés
  - Llista  $\rightarrow |E|$  posicions  $\rightarrow$  mínim un accés memòria versus localització
    - Graf **dens** versus graf **sparse**



Què faríeu servir per codificar tots els enllaços de les pàgines del **www?**

# Algorismes sobre grafs

- Quins vèrtexs són accessibles des de quins?



Ho podem veure com un laberint  
Hem de guardar informació a mida que analitzem/"explorem"  
el graf.

# Algorismes sobre grafs

- Quins vèrtexs són **accessibles** des de quins?

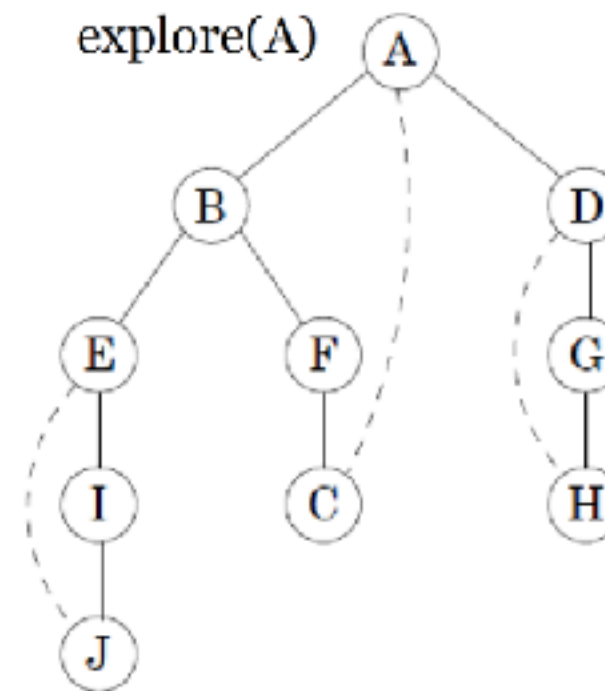
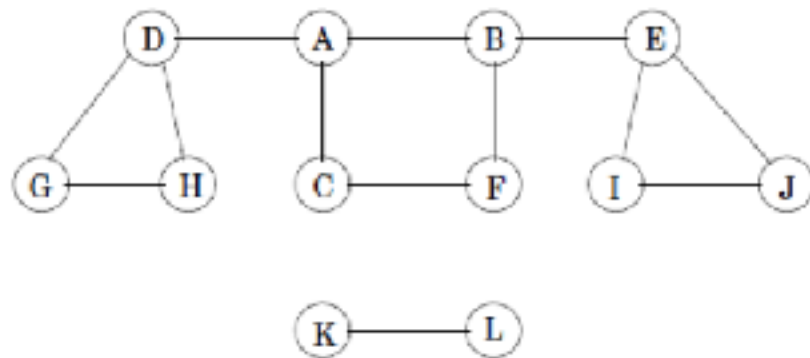
```
procedure explore( $G, v$ )  
Input:    $G = (V, E)$  is a graph;  $v \in V$   
Output:  visited( $u$ ) is set to true for all nodes  $u$  reachable from  $v$   
  
visited( $v$ ) = true  
previsit( $v$ )  
for each edge  $(v, u) \in E$ :  
    if not visited( $u$ ): explore( $u$ )  
postvisit( $v$ )
```

---

# Algorismes sobre grafos

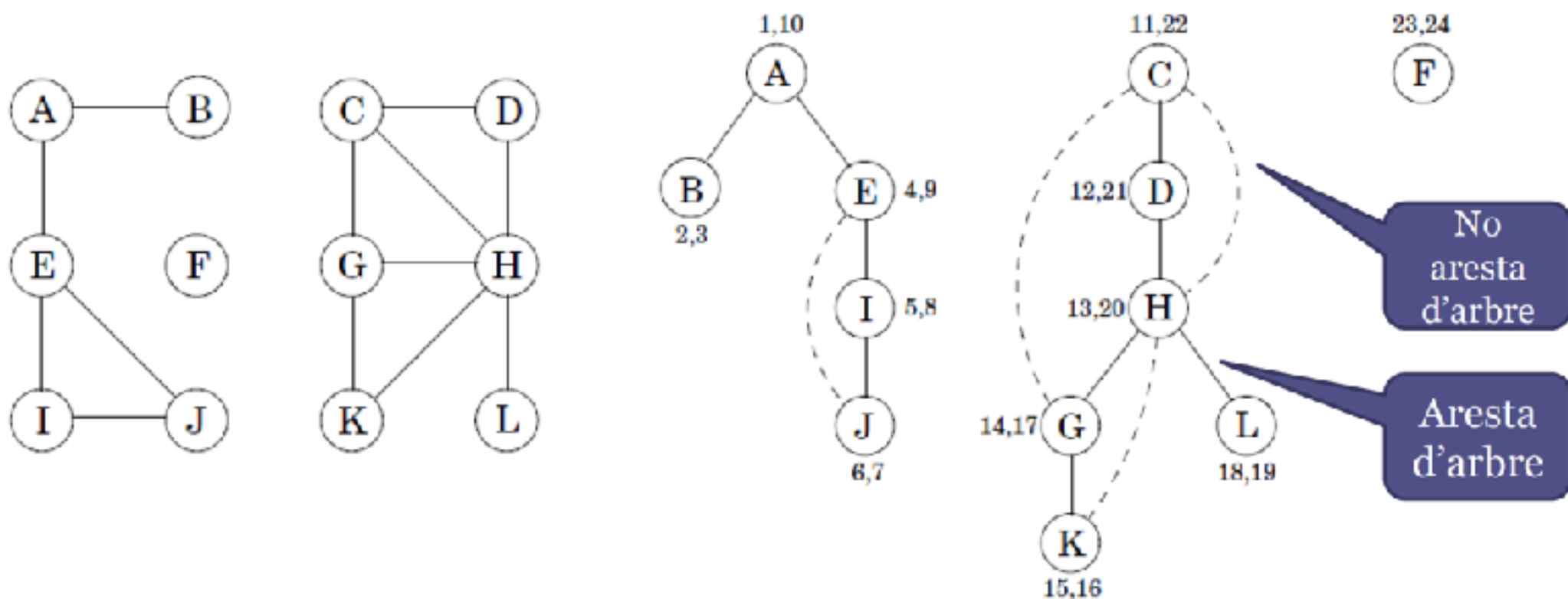
- **Recorregut topològic**
- Busqueda en profunditat ( **Depth-First Search - DFS** )

```
procedure dfs(G)  
  
for all v ∈ V:  
    visited(v) = false  
  
for all v ∈ V:  
    if not visited(v): explore(v)
```



# Algorismes sobre grafs

- **DFS** representa la connectivitat amb un **bosc d'arbres**



- Quina és la complexitat del **DFS**?

Exercici per a casa