

Algorísmica Avançada

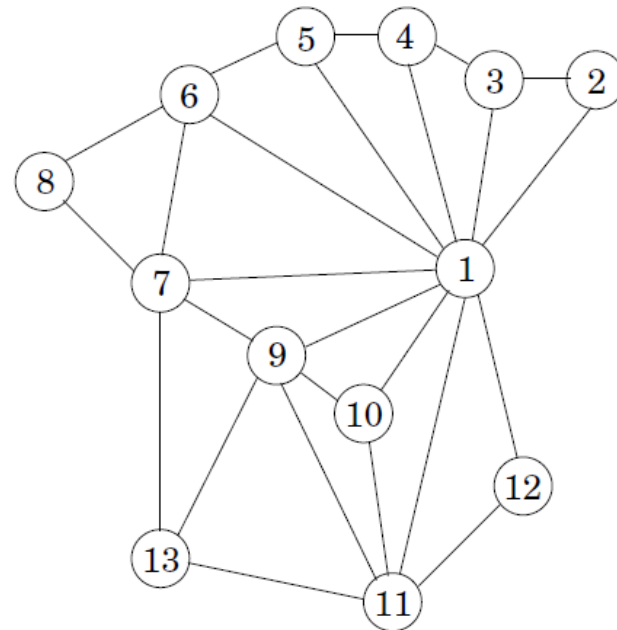
Algorismes sobre grafs I

Sergio Escalera

A series of horizontal lines of varying lengths and colors (teal, light blue, and white) extending from the right side of the slide.

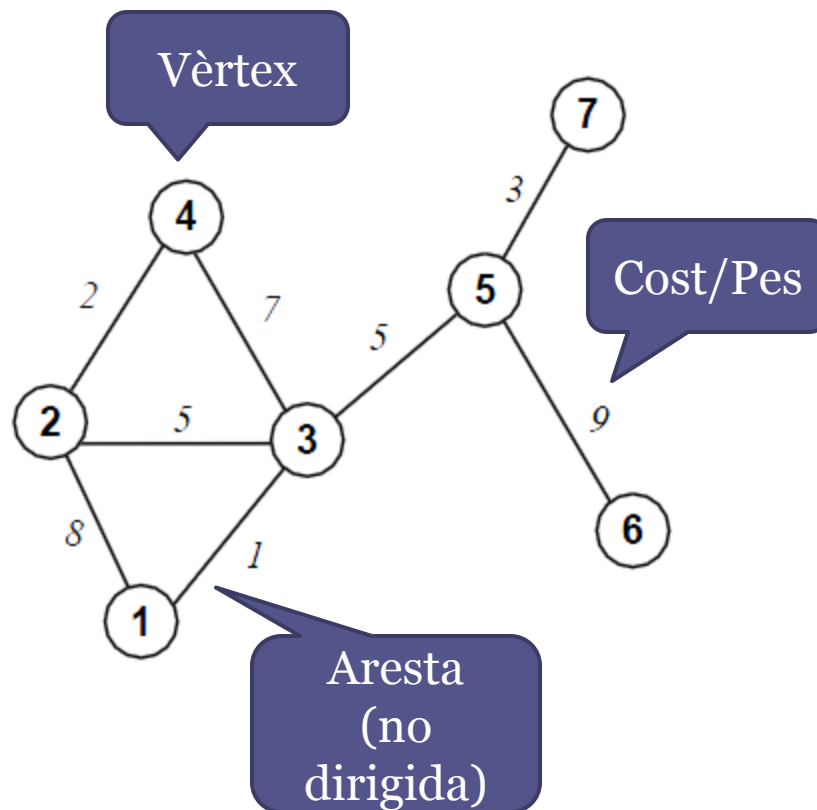
Algorismes sobre grafos

- Per a què serveix un graf?



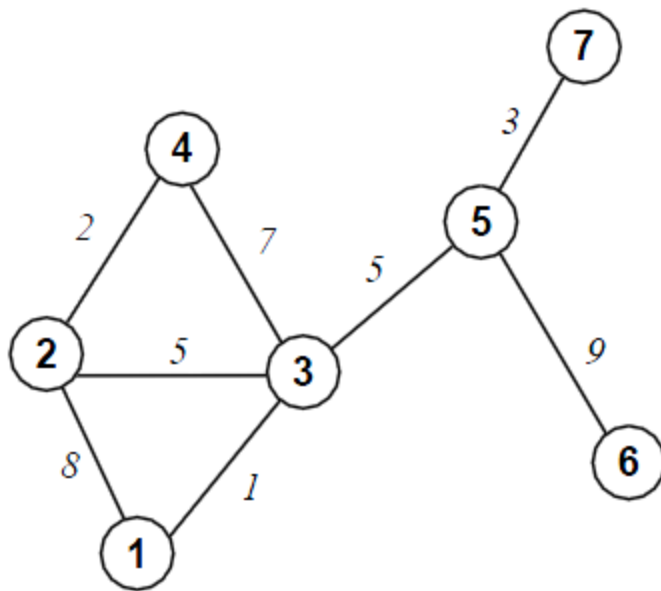
Algorismes sobre grafos

- Com representem un graf? $G=(V,E)$



Algorismes sobre grafos

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

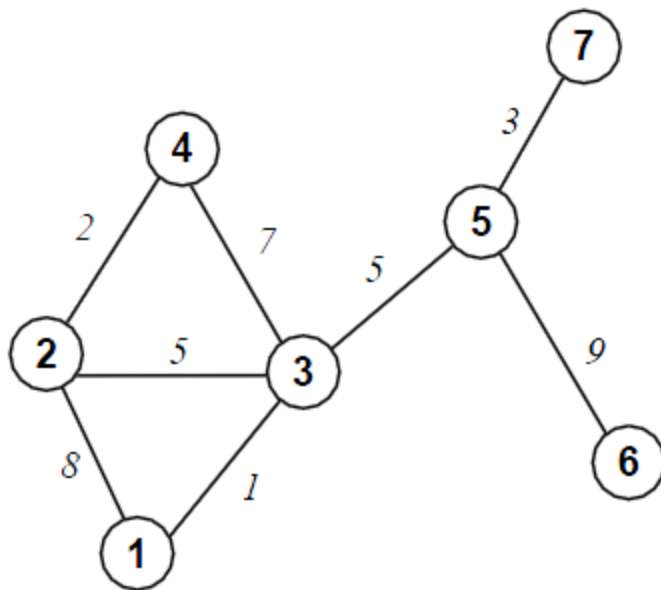


	1	2	3	4	5	6	7
1	0	8	1	∞	∞	∞	∞
2	8	0	5	2	∞	∞	∞
3	1	5	0	7	5	∞	∞
4	∞	2	7	0	∞	∞	∞
5	∞	∞	5	∞	0	9	3
6	∞	∞	∞	∞	9	0	∞
7	∞	∞	∞	∞	3	∞	0

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

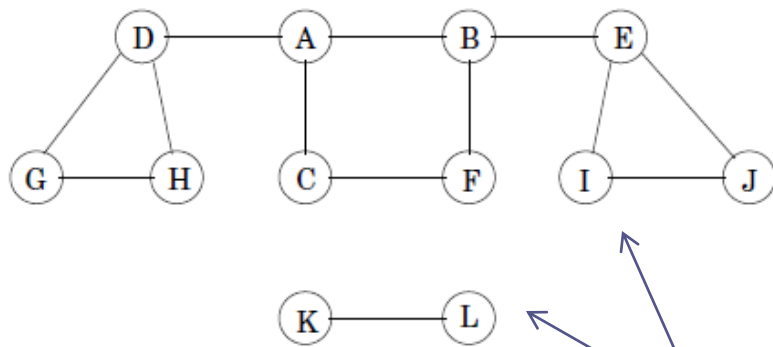
Graf **dens** versus graf **sparse**



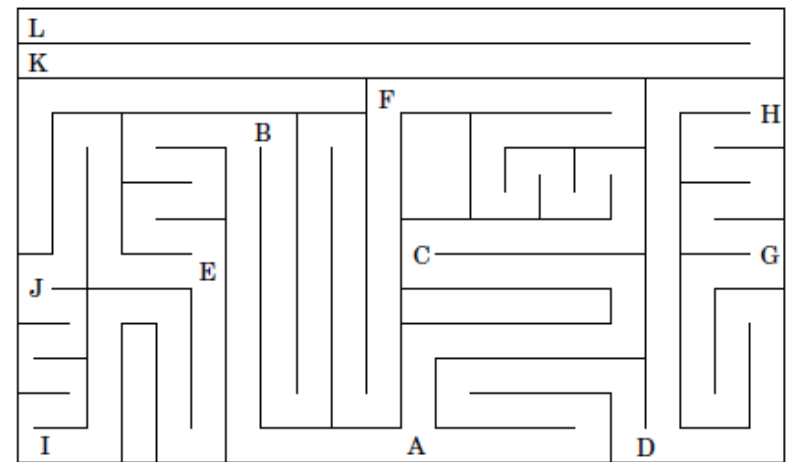
¿què faríeu servir per codificar tots els enllaços de les pàgines del www?

Algorismes sobre grafs

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



Component connex



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)

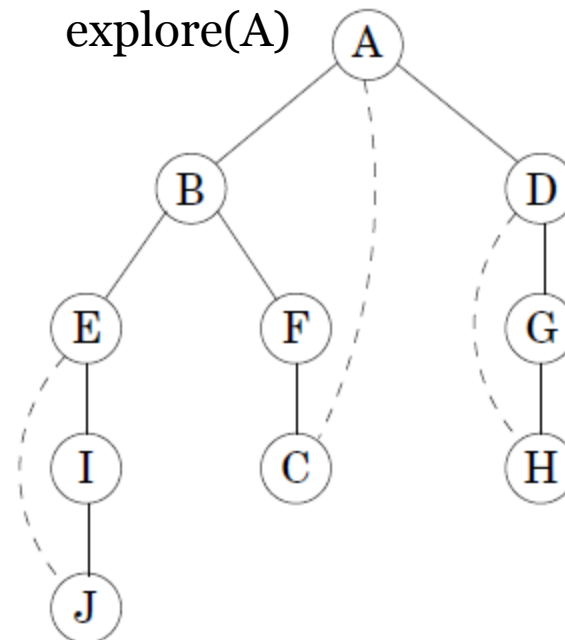
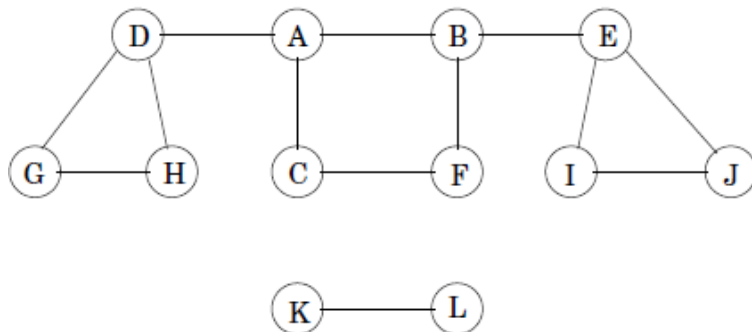
Algorismes sobre grafs

- **Recorregut topològic**
- **Búsqueda en profundidad (Depth-First Search - DFS)**

procedure dfs(G)

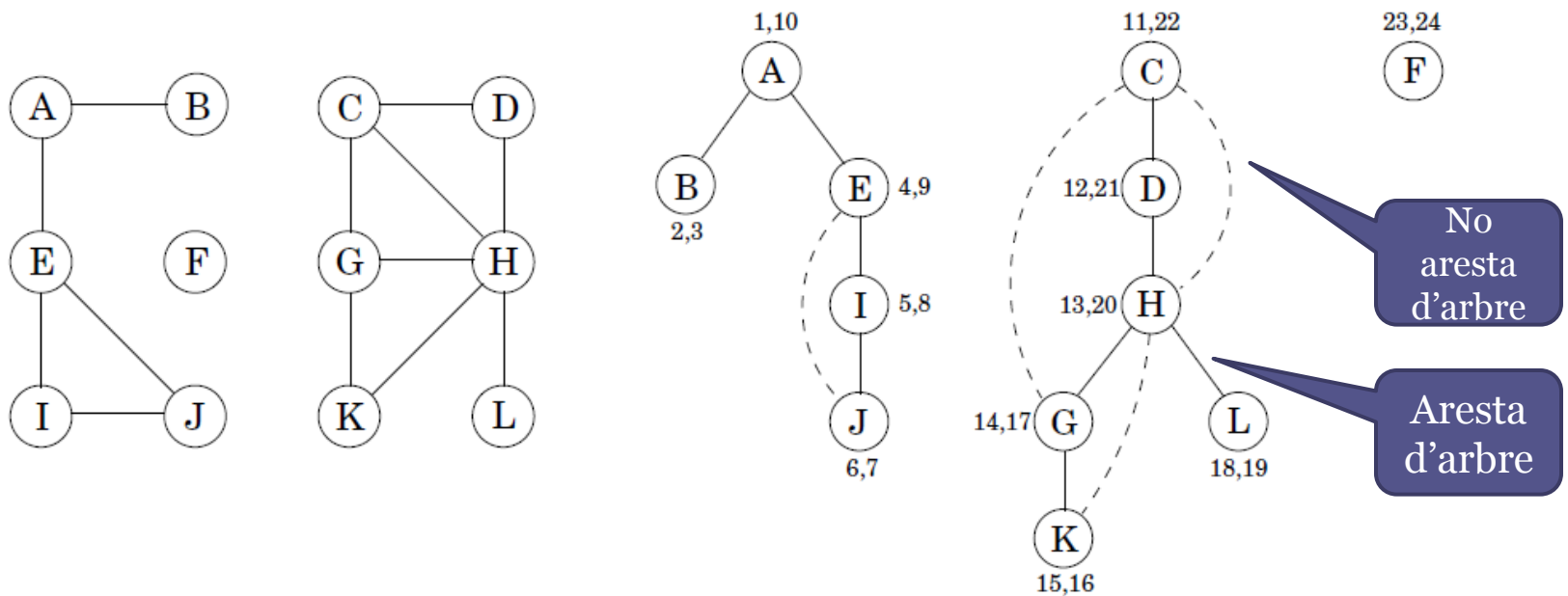
for all $v \in V$:
 $\text{visited}(v) = \text{false}$

for all $v \in V$:
 if not $\text{visited}(v)$: $\text{explore}(v)$



Algorismes sobre grafs

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



- Complexitat de DFS?

Algorismes sobre grafs

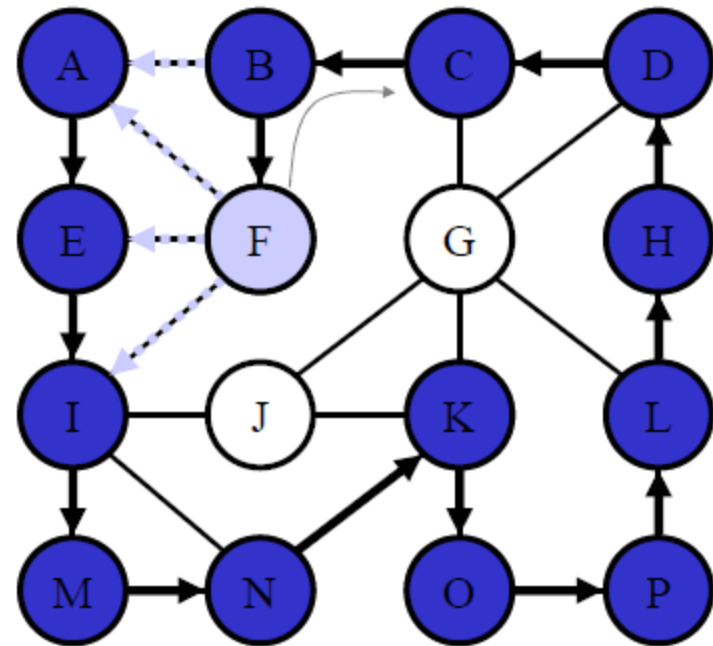
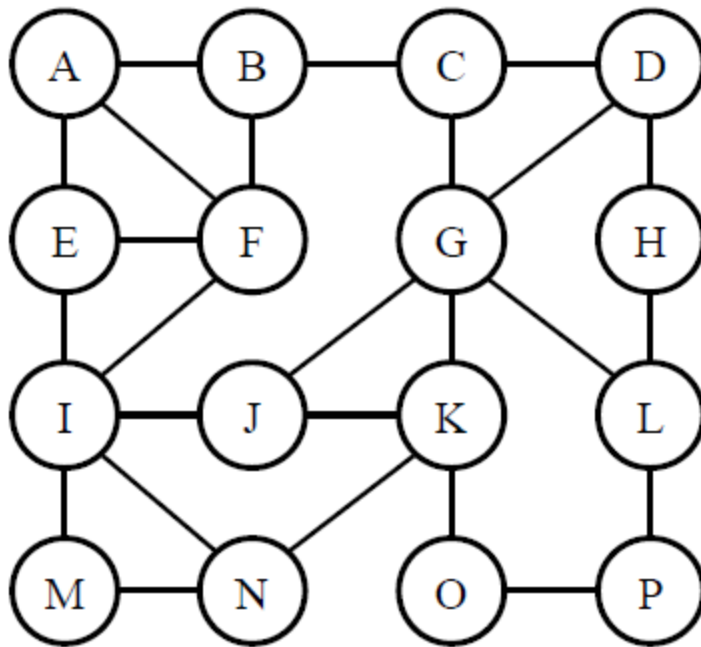
- Complexitat de DFS?

Teorema: la suma de tots els graus de tots els nodes és igual a 2 cops el nombre d'arestes del graf.

$$\Theta(|V| + |E|) \rightarrow \Theta(2|E|) \rightarrow \Theta(|E|)$$

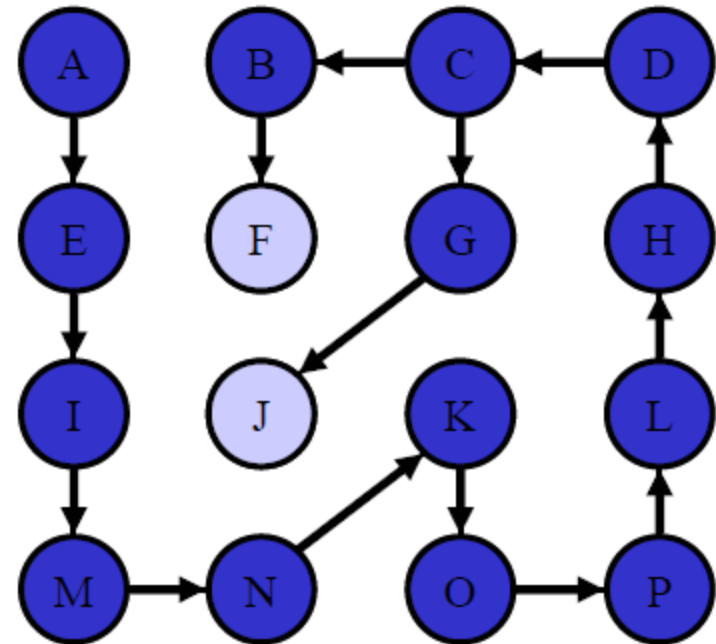
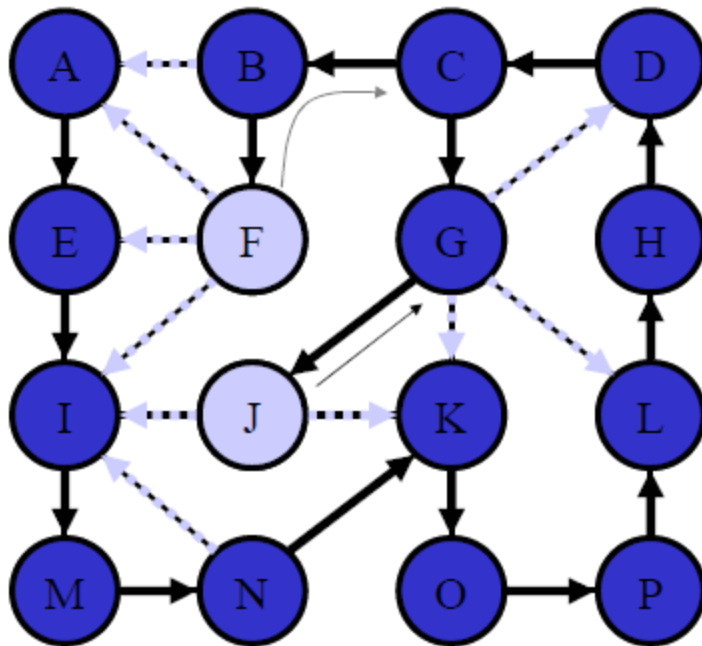
Algoritmos sobre grafos

- Example



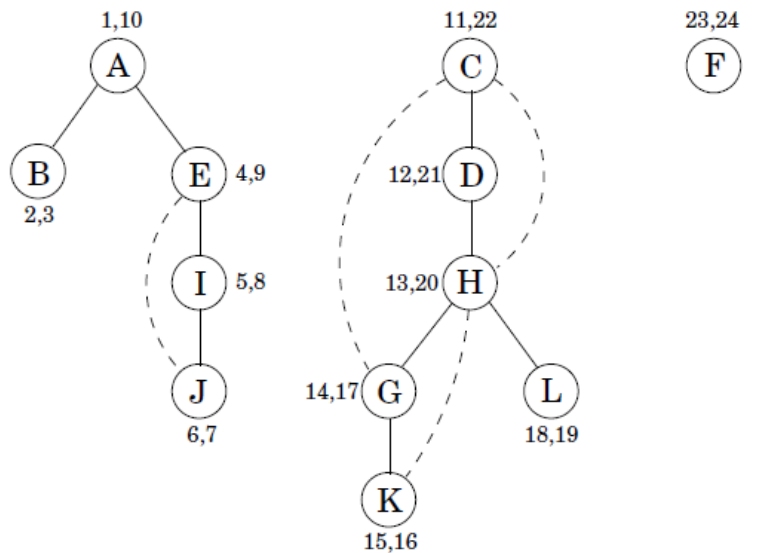
Algoritmes sobre grafos

- Exemple



Algorismes sobre grafs

- DFS també ens soluciona un altre problema de grafs: els components conexas



$\{A, B, E, I, J\}$ $\{C, D, G, H, K, L\}$ $\{F\}$

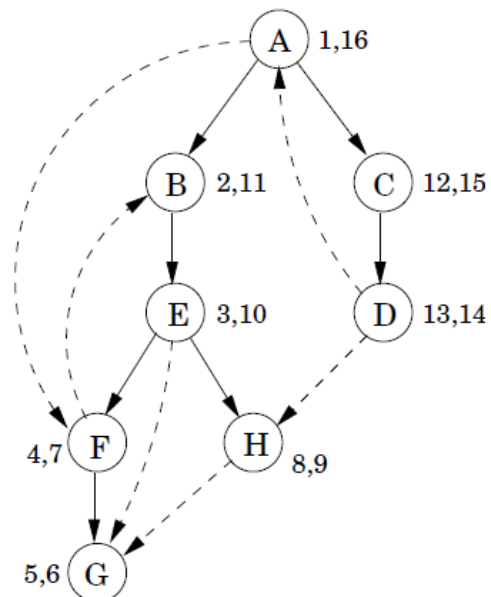
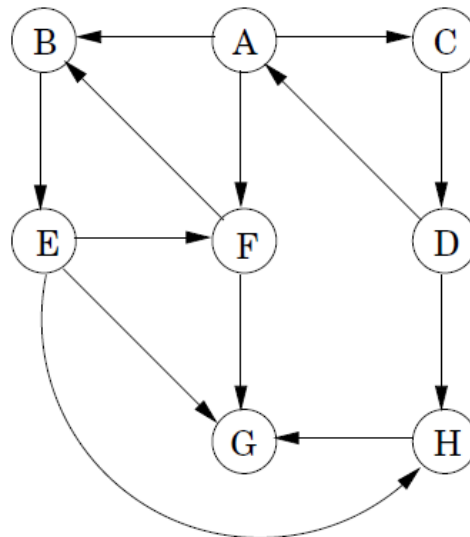
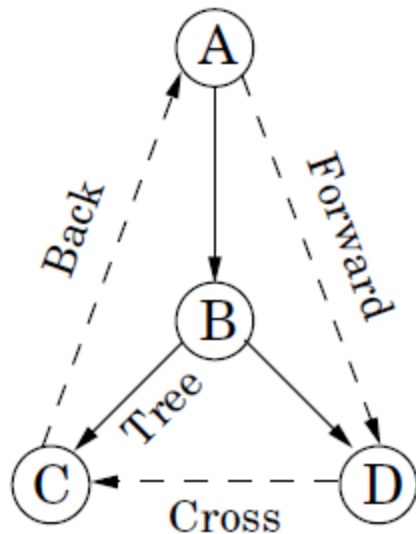
Cada crida a explore crea un nou arbre

I es troba una **nova component conexa**

Algorismes sobre grafs

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

DFS tree



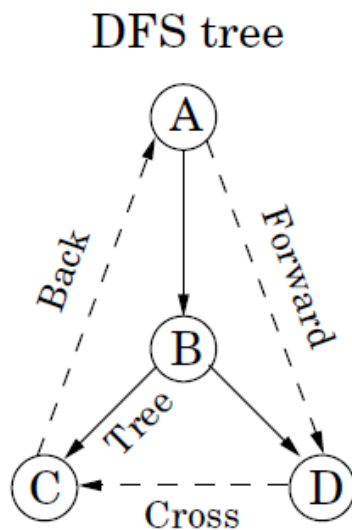
Algorismes sobre grafs

- En els grafs **no dirigits**, una **component connexa** conté com a mínim un **cicle**
- En els grafs **dirigits**, un **cicle** ha de començar i acabar en el mateix vèrtex existint connectivitat, sinó és **acíclic**

$$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_k \rightarrow v_0$$

Algorismes sobre grafs

- Cicles en grafs dirigits
- Podem trobar un cicle en un graf dirigit en temps lineal? → Sí



Hem vist que DFS té complexitat lineal

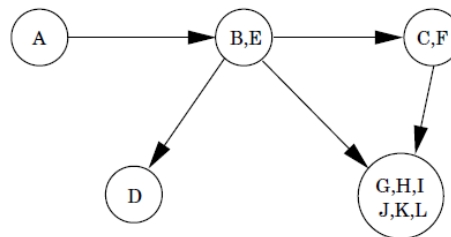
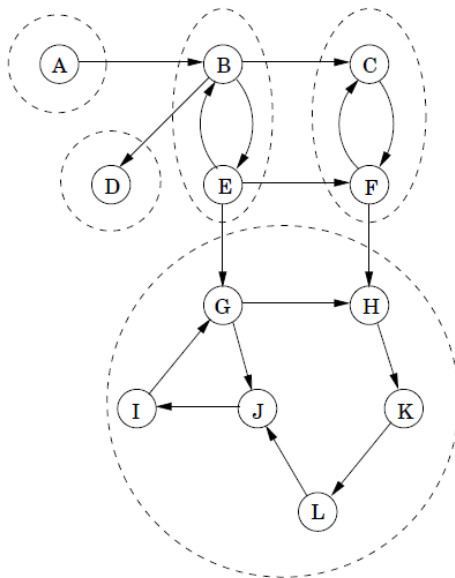
Propietat: Un graf dirigit té un cicle si i només si l'algorisme DFS troba una aresta “back”.

Algorismes sobre grafs

- Els grafs dirigits acíclics són molt comuns:
- Nosaltres modelem ó “intentem” modelar les nostres tasques quotidianes en un ordre determinat, una rere l’altre.
 - Els grafs acíclics modelen relacions com jerarquies o dependències temporals

Algorismes sobre grafs

- Connectivitat en grafs dirigits
- Hi ha d'haver connectivitat $u \rightarrow v$ i $v \rightarrow u$
- **Components forts connexes**

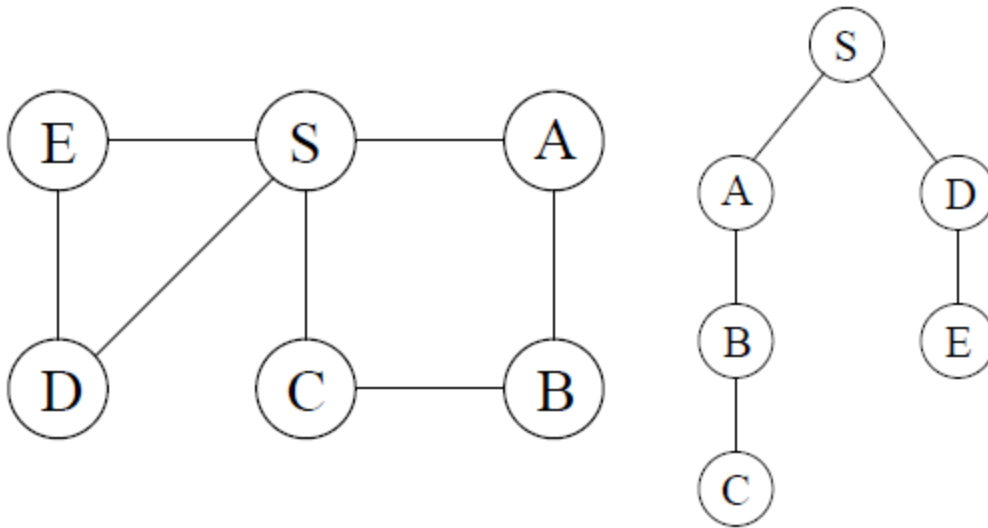


També els podem trobar amb complexitat lineal fent ús de l'algorisme DFS

Algorismes sobre grafs

- Fins ara hem parlat de connectivitat, però no hem analitzat el cost del camí que hem trobat entre els punts connectats.
- **DFS assegura el camí més curt entre 2 punts connectats en un graf no dirigit???**

Algorismes sobre grafs



Podem definir el camí entre 2 punts com el número d'arestes fins arribar, o el número de vèrtexs que travessem, o la suma dels pesos de les arestes, dels vèrtexs, etc.

Algoritmos sobre grafos

- Cerca en amplitud (Breadth-first search)

procedure `bfs`(G, s)

Input: Graph $G = (V, E)$, directed or undirected; vertex $s \in V$

Output: For all vertices u reachable from s , `dist`(u) is set to the distance from s to u .

for all $u \in V$:
 `dist`(u) = ∞

`dist`(s) = 0

$Q = [s]$ (queue containing just s)

while Q is not empty:

$u = \text{eject}(Q)$

 for all edges $(u, v) \in E$:

 if `dist`(v) = ∞ :

`inject`(Q, v)

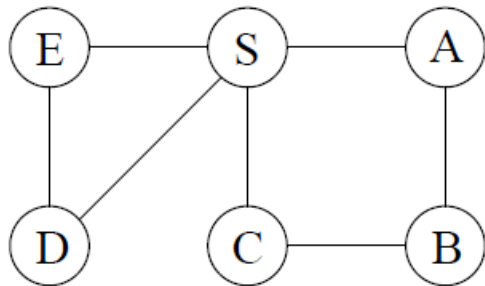
`dist`(v) = `dist`(u) + 1

Algorismes sobre grafs

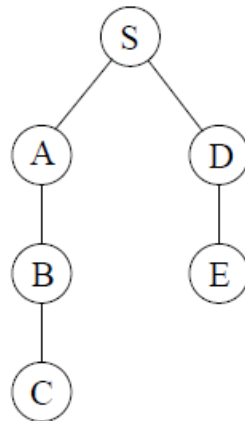
- **BFS** té un codi similar a **DFS**, però fa ús d'una **cua** en lloc d'una **pila**.
- Els arbres generats per BFS es diuen **arbres de camí mínim**.
- Si fem ús correcte del “cost del camí” a l'algorisme BFS trobem el camí mínim d'un vèrtex a la resta de vèrtex dins d'un graf!

Algoritmos sobre grafos

Graf

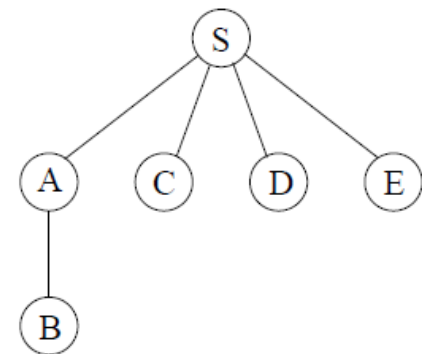


DFS



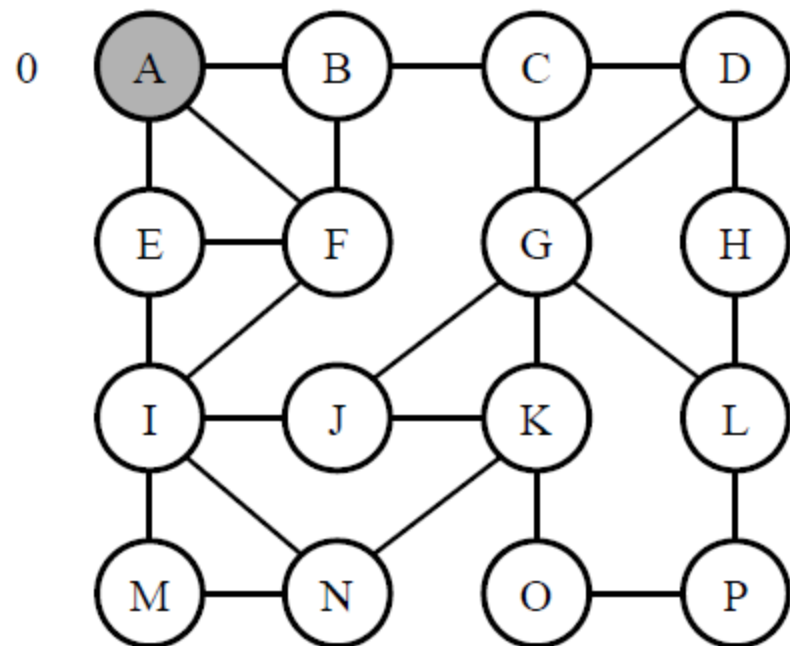
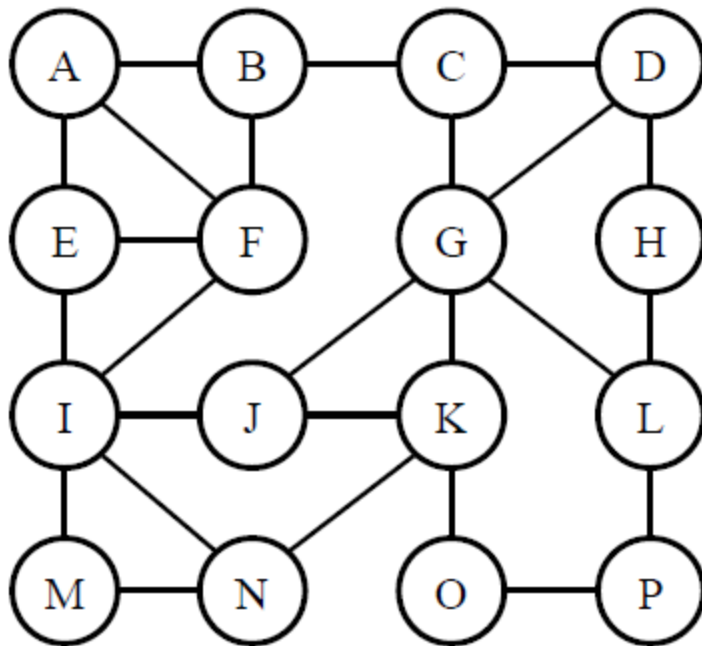
BFS

Order of visitation	Queue contents after processing node
<i>S</i>	[<i>S</i>]
<i>A</i>	[<i>A C D E</i>]
<i>C</i>	[<i>C D E B</i>]
<i>D</i>	[<i>D E B</i>]
<i>E</i>	[<i>E B</i>]
<i>B</i>	[<i>B</i>]
	[]



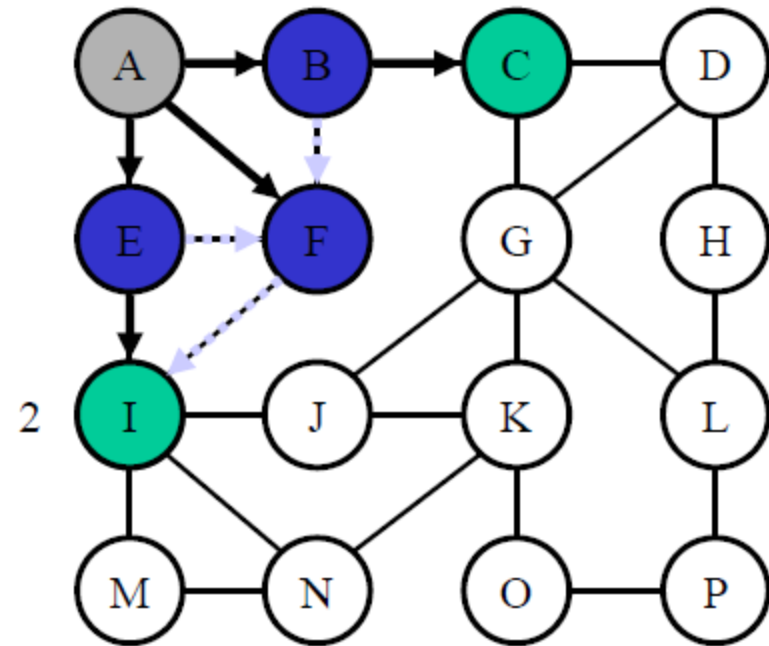
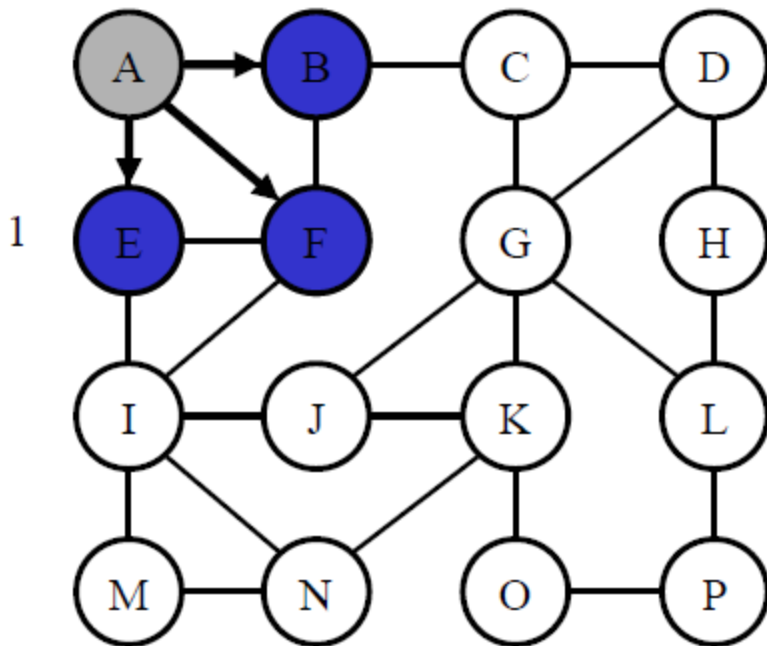
Algoritmos sobre grafos

Exemple



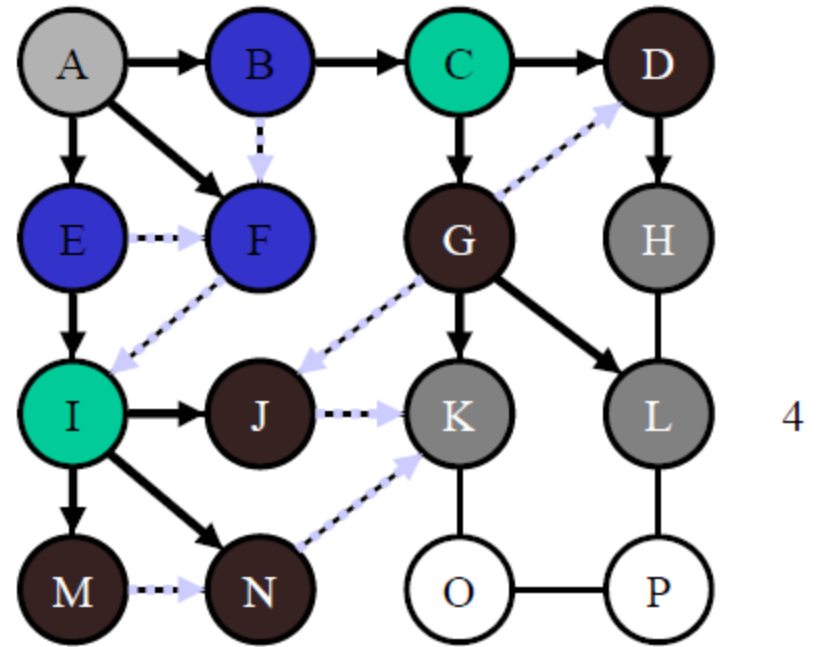
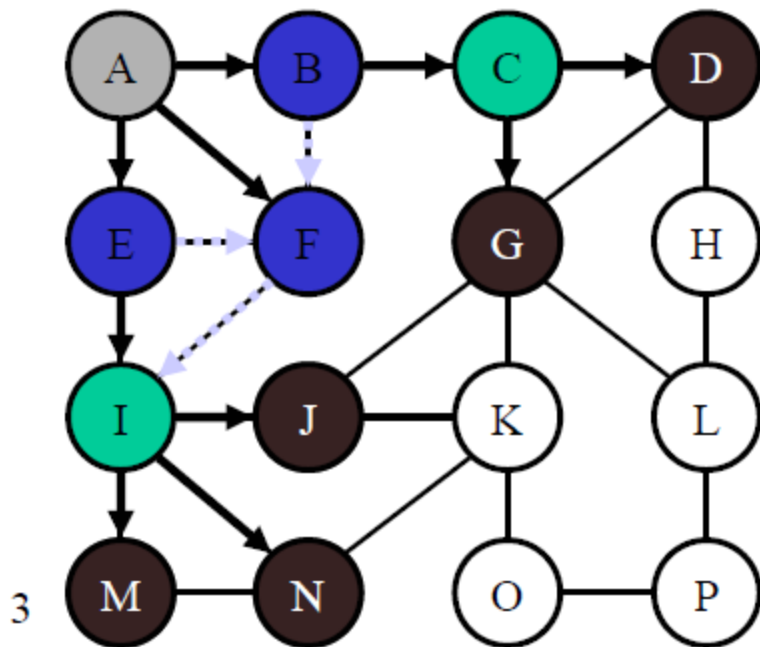
Algoritmes sobre grafos

Exemple



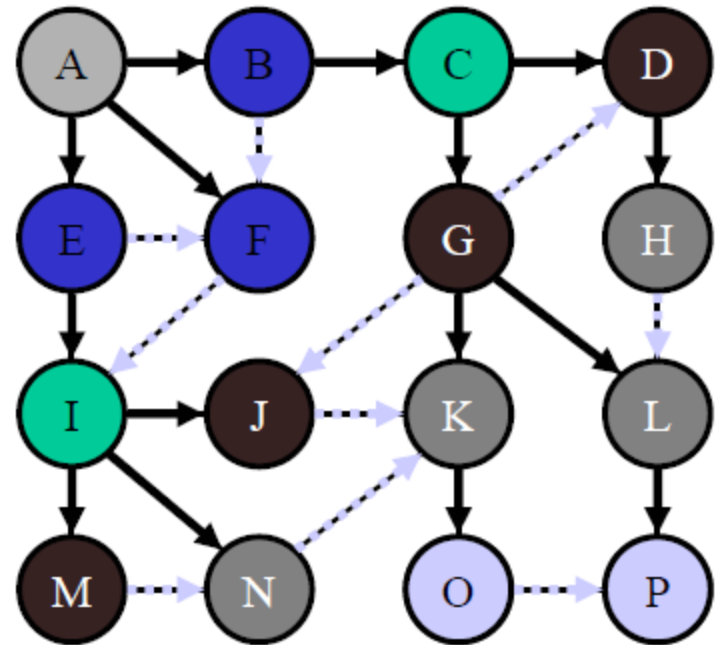
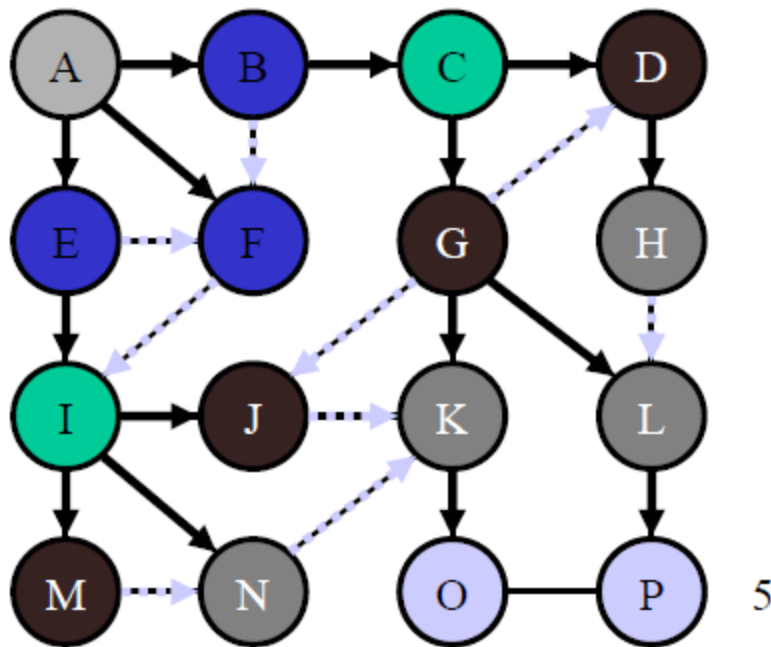
Algoritmes sobre grafos

Exemple



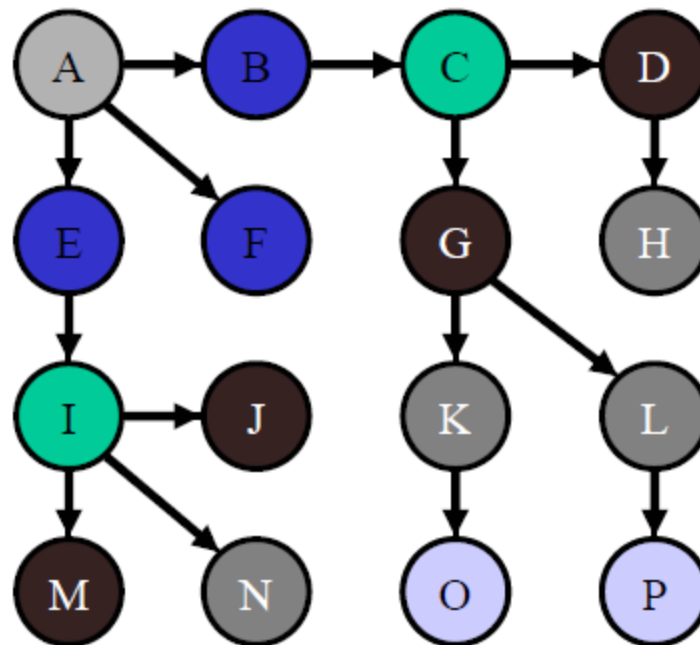
Algoritmos sobre grafos

Exemple



Algoritmos sobre grafos

Exemple



Algorismes sobre grafs

- BFS versus DFS
- Una altra diferència és que BFS només té en compte els nodes que estan connectats a un node s, els altres són ignorats
 - → només es genera un arbre de camins mínims