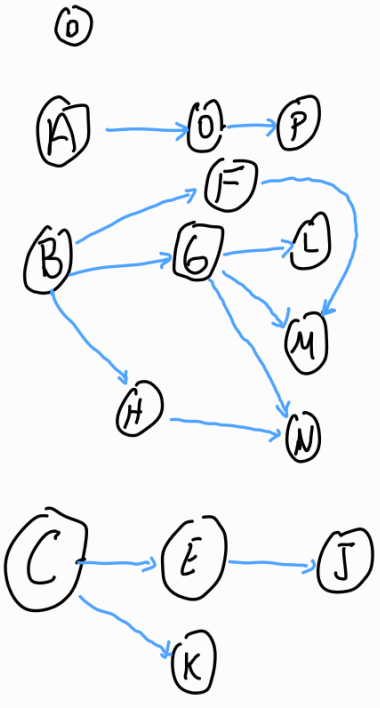
Aplicaciones DFS / BFS.

Topological Sort. (Grafos Dirigidos)

Lista (ordenamiento) de los nodos tal que para todo arco (x,y), x aparece antes que y en la lista.

$(x,y) \rightarrow$ 'X es pre requisito de Y

A → Introducción a ing. sistemas. ✓
B → Calculo 1 ✓
C → Programación 1 ✓
D → Pensamiento sistemico.
E → POO ✓
F → Algebra Lineal ✓
G → Calculo 2 ✓
H → Fisica 1 ✓
J → Estructuras de datos ✓
K → Bases de datos
L → Calculo 3 ✓
M → Prob y estadistica.
N → Fisica 2 ✓
O → Elementos
P → Arquitectura de PC



G, L, B. ✗

N, H, G ✗

A, C, B, D, G, K, H ✓

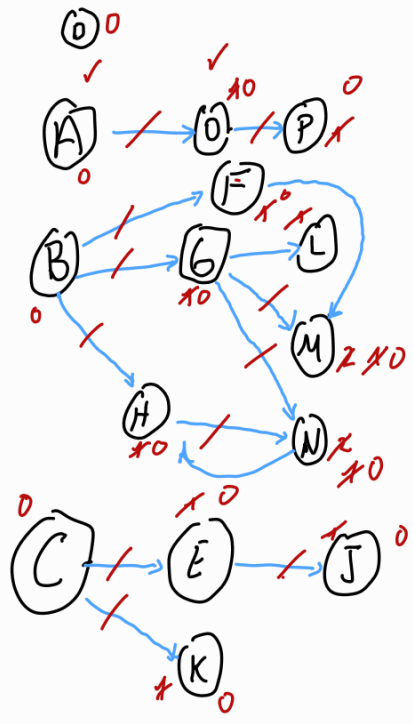
Grafos dirigidos

IN-degree → # de aristas que entran en el nodo

Out-degree → # de aristas que salen del nodo.

BFS

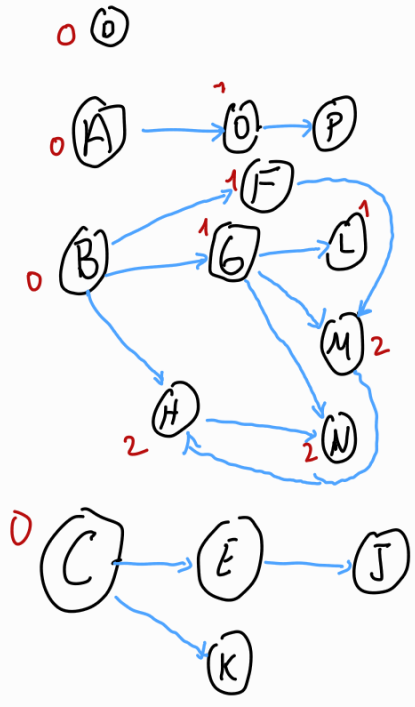
↳ IN-degree



ListaMaterias:

[D, A, B, C, O, F, G, H, E, K, P, L, N, M]

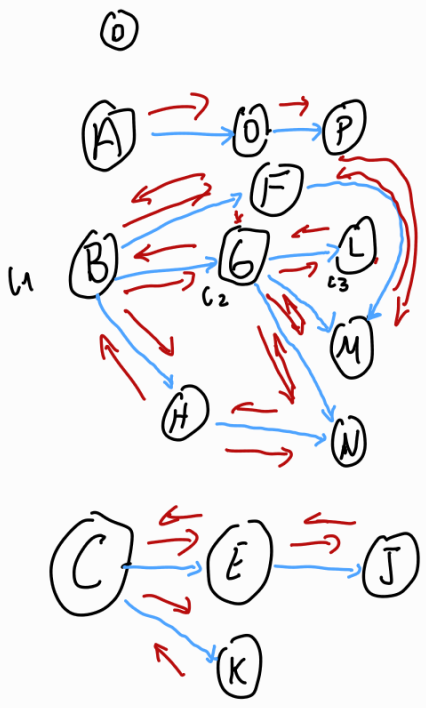
J J



Tarea. 1. Que pasa en la estrategia basada en BFS para calcular el TopSort cuando hay ciclos.

¿Cómo identificar ciclos con esta estrategia?

Basada en DFS



Cada vez que termine la recursión (no haya mas pasos recursivos por hacer) inserto el elemento en una lista

topsort = [L, M, N, G, H, F, B, P, O, A, D, J, E, K, C]

Reverse al topsort

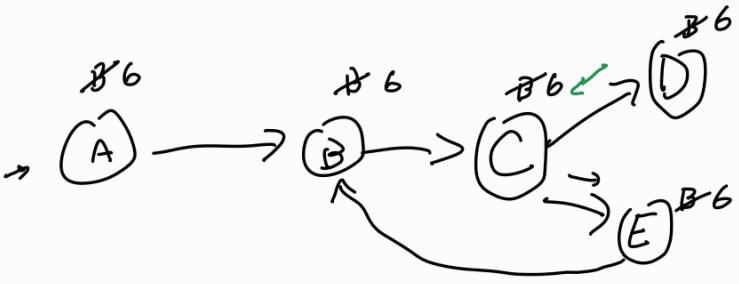
1 2 2 3 3 1
topsort = [C, K, E, J, D, A, O, P, B, F, H, G, N, M, L]

Stack recursivo.

~~B~~ ~~G~~ ~~L~~ ~~M~~ ~~N~~ ~~H~~ ~~F~~ ~~A~~ ~~O~~ ~~P~~ ~~D~~ ~~C~~ ~~E~~ ~~J~~ ~~K~~

Color de los nodos en una dFS.

Blanco → Esta pendiente por explorar
Gris → Está dentro del stack recursivo (tiene cosas pendientes)
Negro → Salió del stack recursivo (visitado.



Stack
A
B
C
~~D~~
E

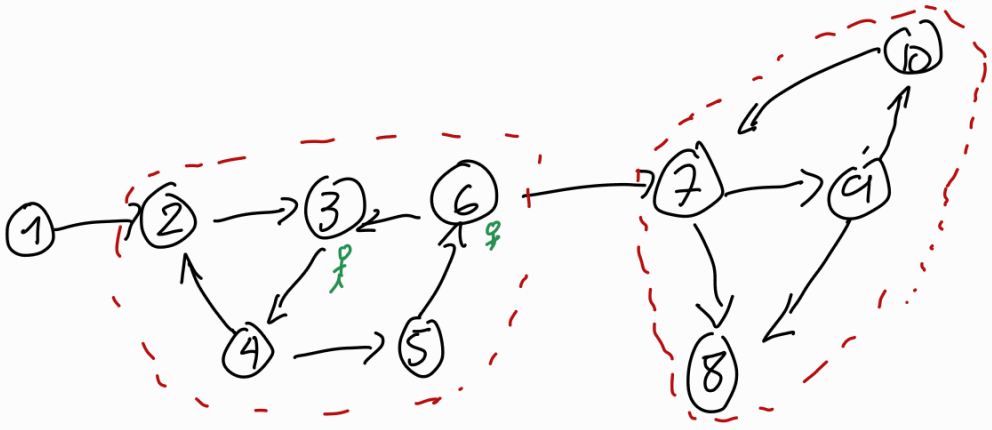
1. Si de un nodo gris llego aun nodo gris
   ↳ hay un ciclo.

Lista = [D

Tarea. ¿Si de un nodo gris llego a un nodo negro significa que hay ciclo?

Componente fuertemente conexa. (Grafos dirigidos)

un conjunto de nodos $(N_0, N_1, N_2 \dots, N_n)$ t.q para cualquier pareja N_i, N_j puedo llegar desde N_i hasta N_j



Varias dfs.