TP série n°1: test de connexité des réseaux «critiques»

I- Problème

Lors de la conception et surtout du test d'un réseau de communication dans les domaines *critiques* (sécurité, défense, finance…), il est primordial, pour l'ingénieur chargé de la conception du réseau de s'assurer que la configuration utilisée garantit, pour certaines parties du réseau, une communication depuis n'importe quel nœud du réseau vers n'importe quel autre.

II-Modèle de graphe

Un moyen de résoudre ce problème est de commencer par modéliser le réseau à l'aide d'un **graphe orienté** G=(S,A) où:

- chaque sommet $s \in S$ représente un **nœud** du réseau,
- chaque arc $(s_i, s_j) \in A$ représente une **connexion** entre le nœud représenté par son extrémité initiale s_i et le nœud représenté par son extrémité terminale s_j

III-Solution

Le problème de l'analyse de la connexité des sous-réseaux critiques est alors posé en termes de connexité du modèle de graphe. Plus précisément, il est ramené au problème classique de recherche, dans un graphe, de **composantes fortement connexes**.

Dans cette perspective, il est proposé un algorithme (Kosaraju-Sharir) qui recherche toutes les **composante fortement connexe** d'un graphe.

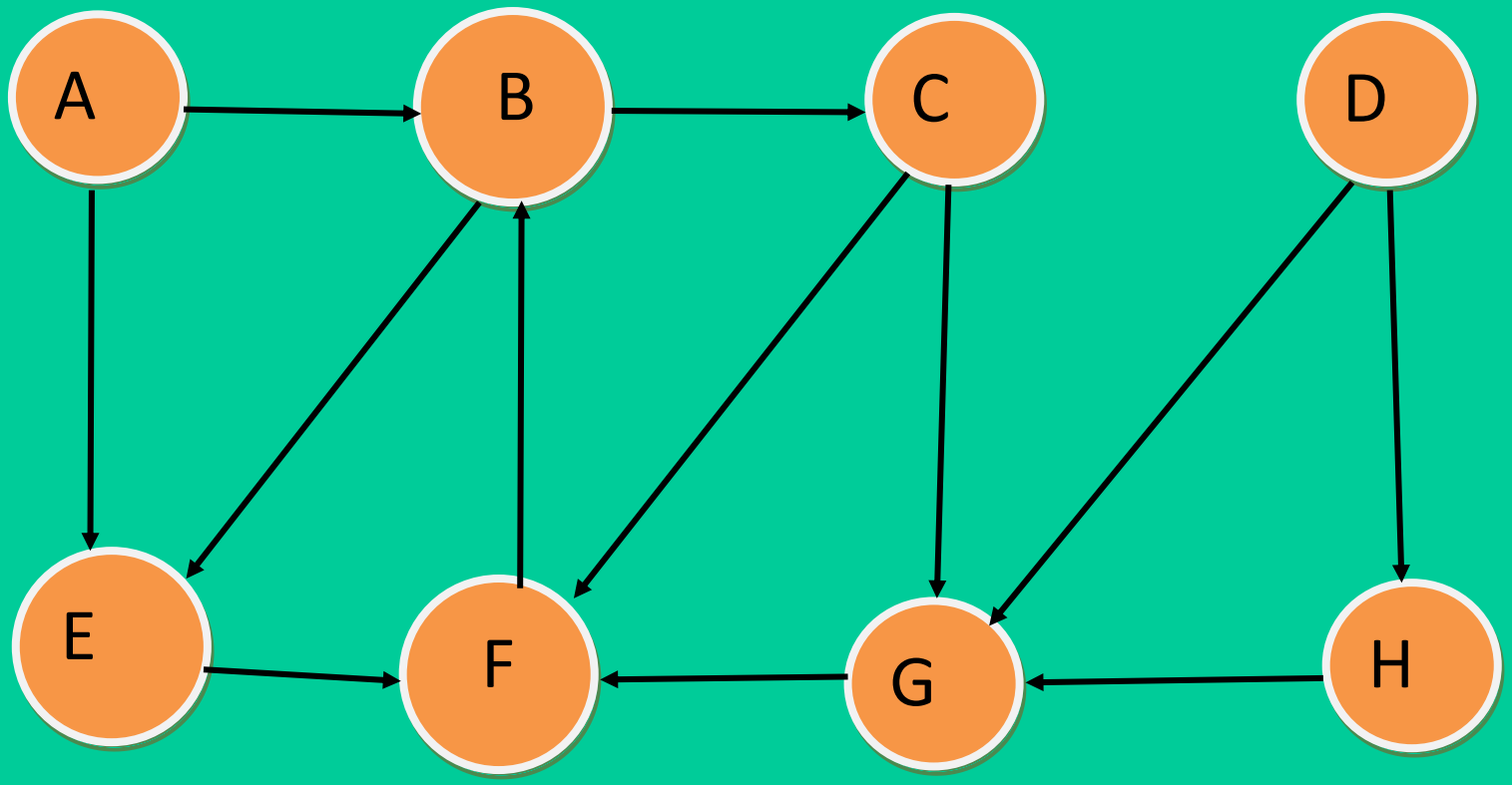
Pour le problème énoncé, l'algorithme fonctionne en deux étapes :

-il lance un premier **parcours en profondeur** sur *G* pour obtenir une liste LISTE des sommets de *G* triée dans l'ordre **décroissant** par rapport à la **date fin**.

- il lance, ensuite, un second parcours en profondeur sur le **graphe dual** $G^t(S,A^t)$ avec, dans chaque la boucle de parcours, un marquage des sommets dans l'ordre spécifié par LISTE.

On montrera que la **complexité** de cet algorithme est en $O(|S| + |A|)$.

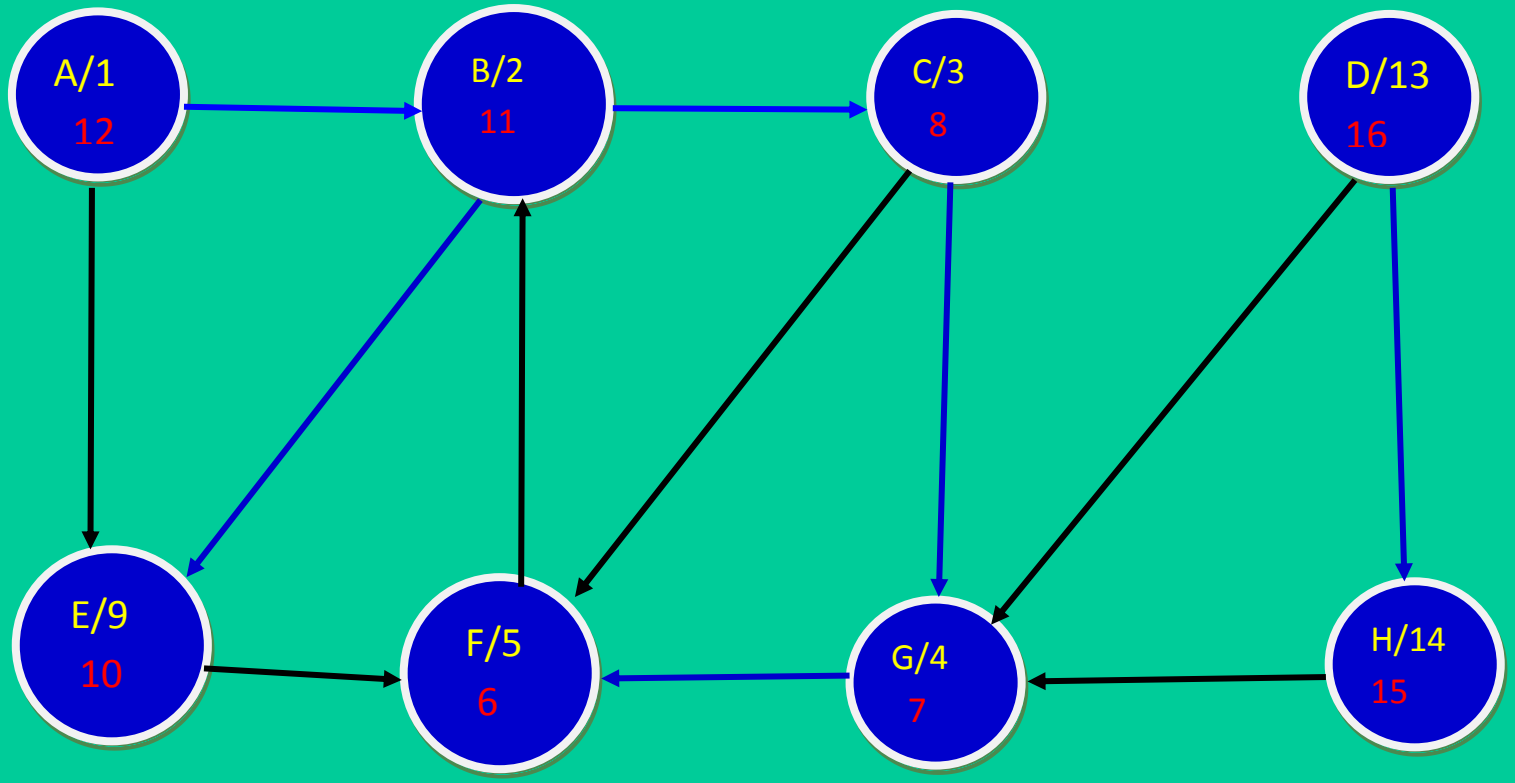
Application de l'algorithme



Trie dans l'ordre date début / date fin

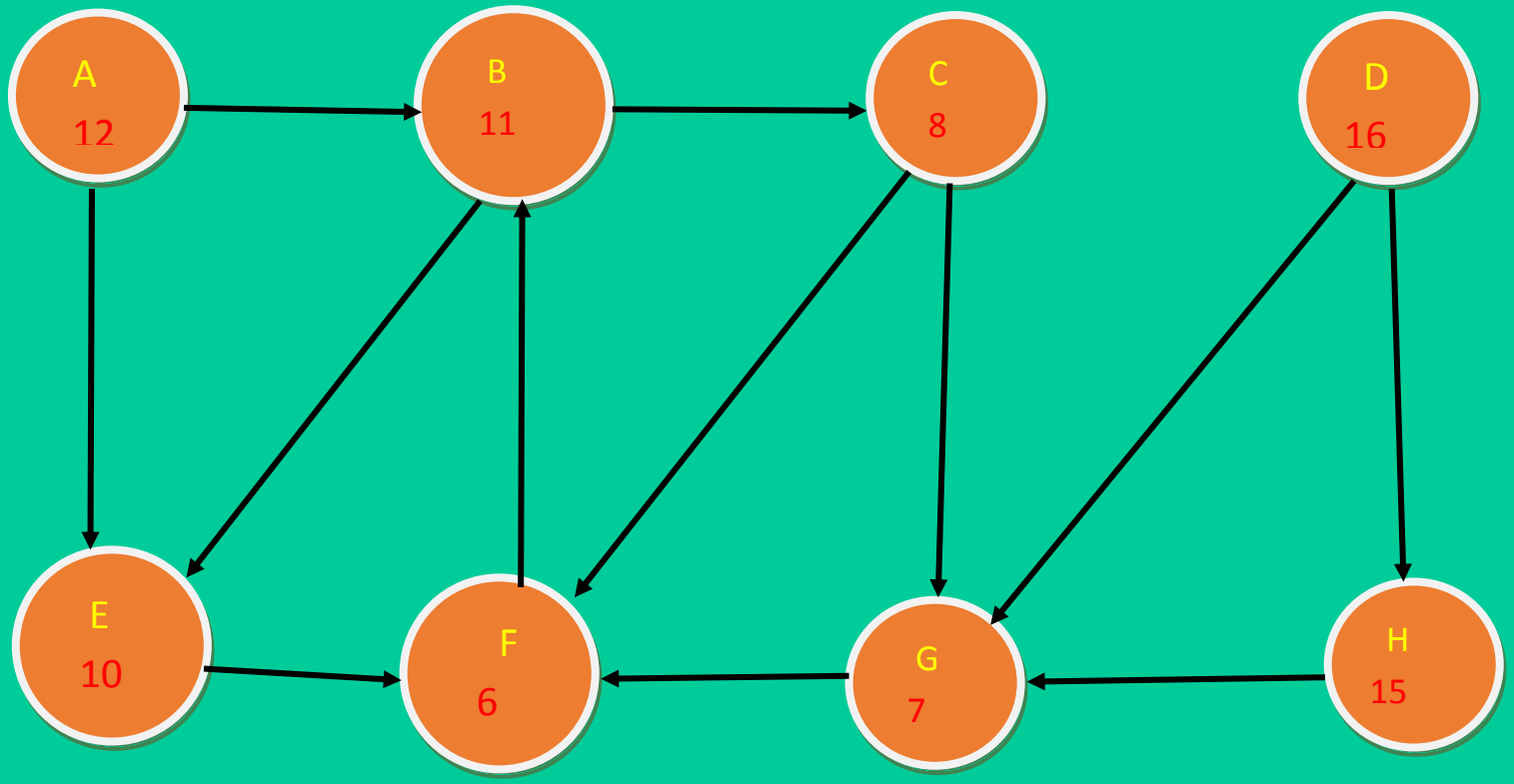
Date fin (en rouge) : **6,7,8,10,11,12,15,16**

Date début (en jaune) : **1,2,4,5,9,13,14**

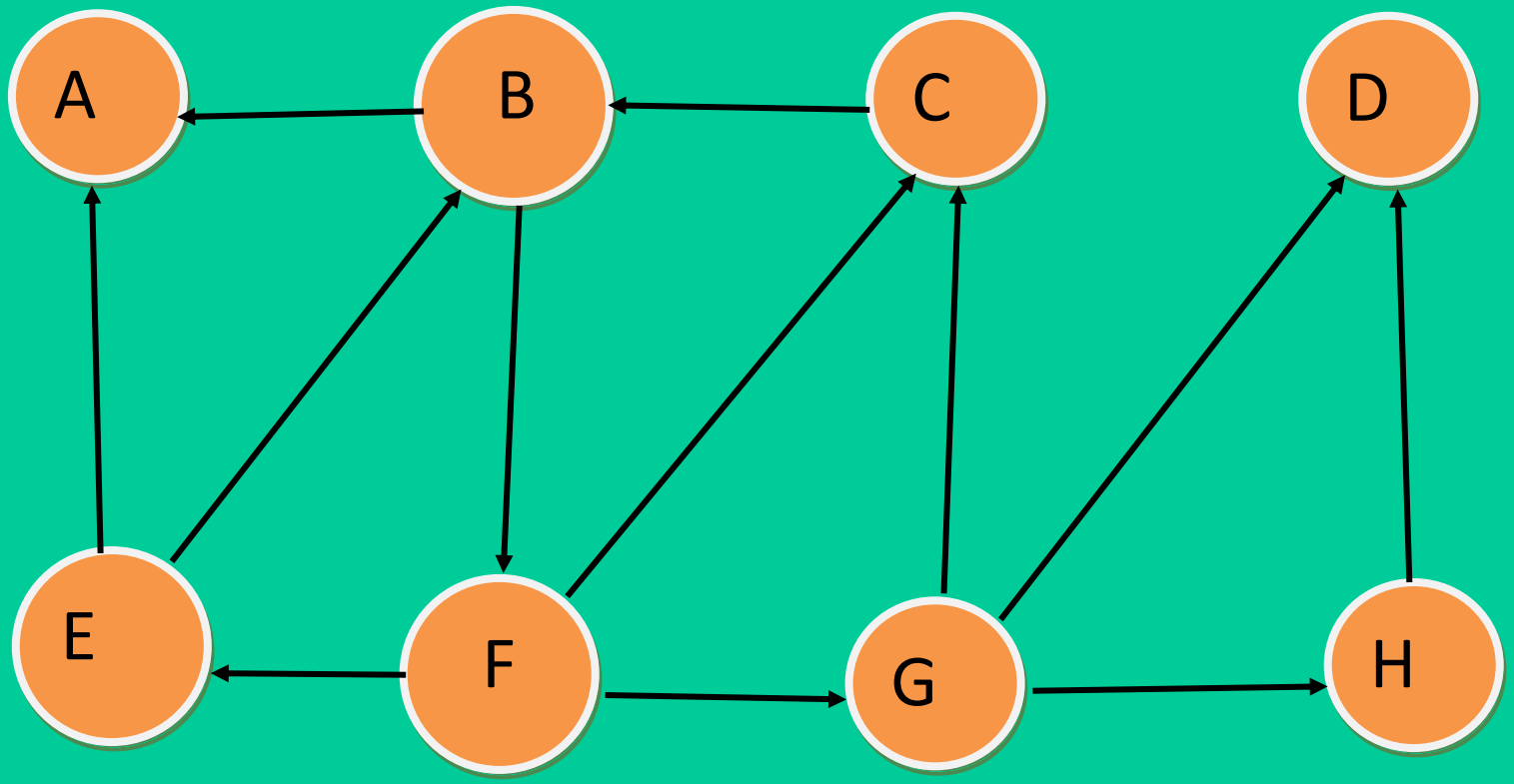


Trie dans l'ordre décroissant de date fin

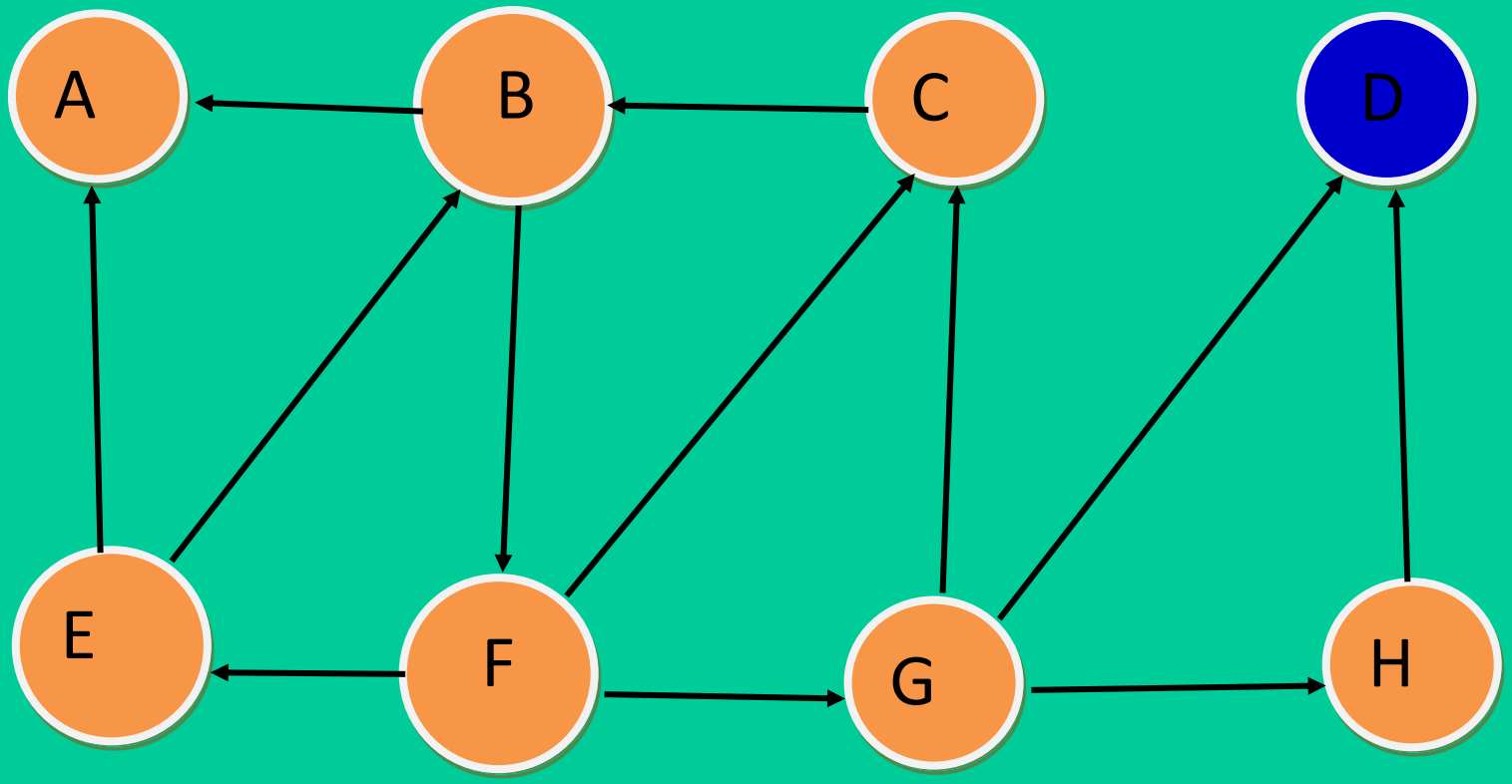
LISTE = **[D, H, A, B, E, C, G, F]**



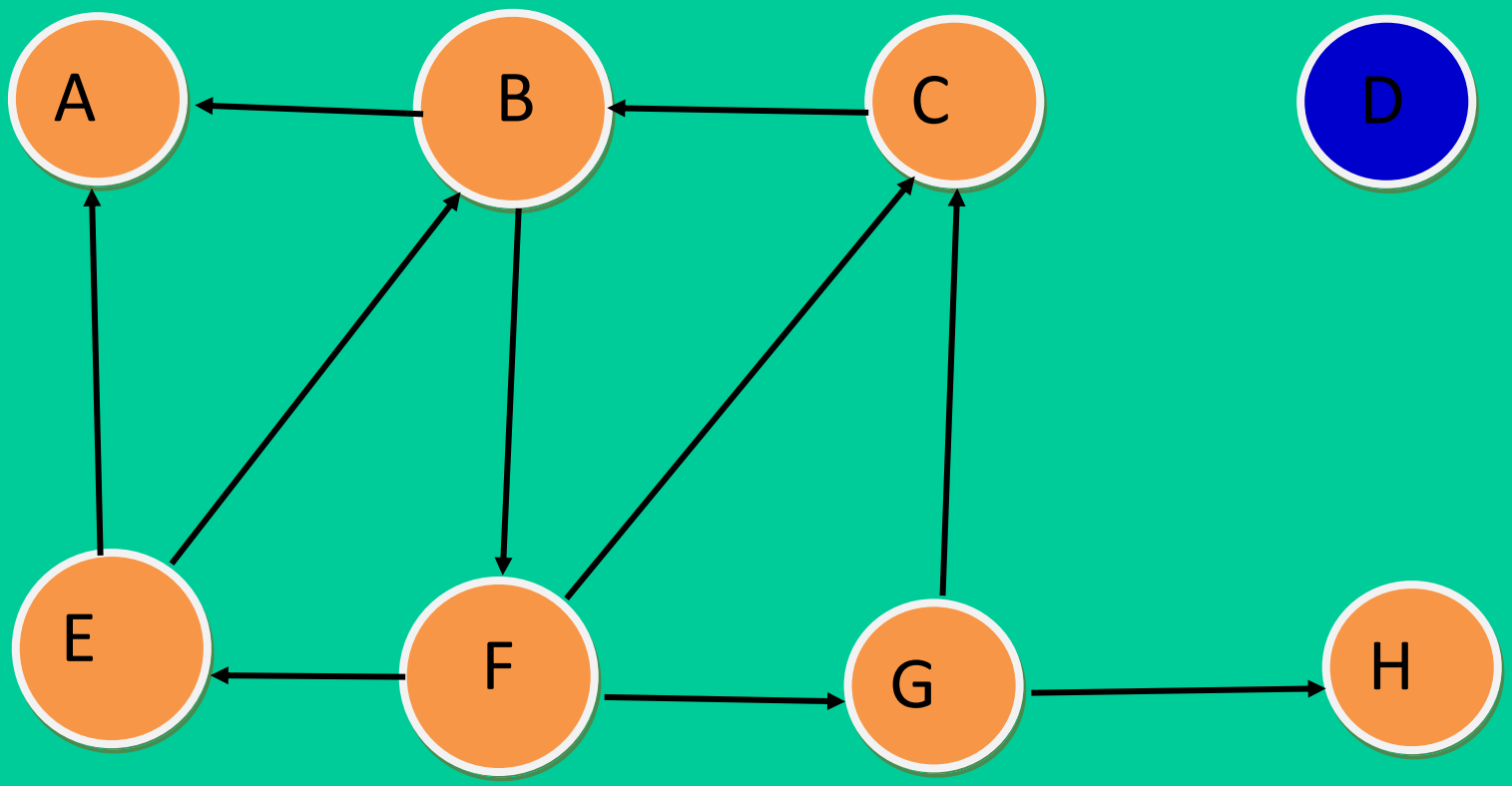
On construit le **graphe dual** G^t



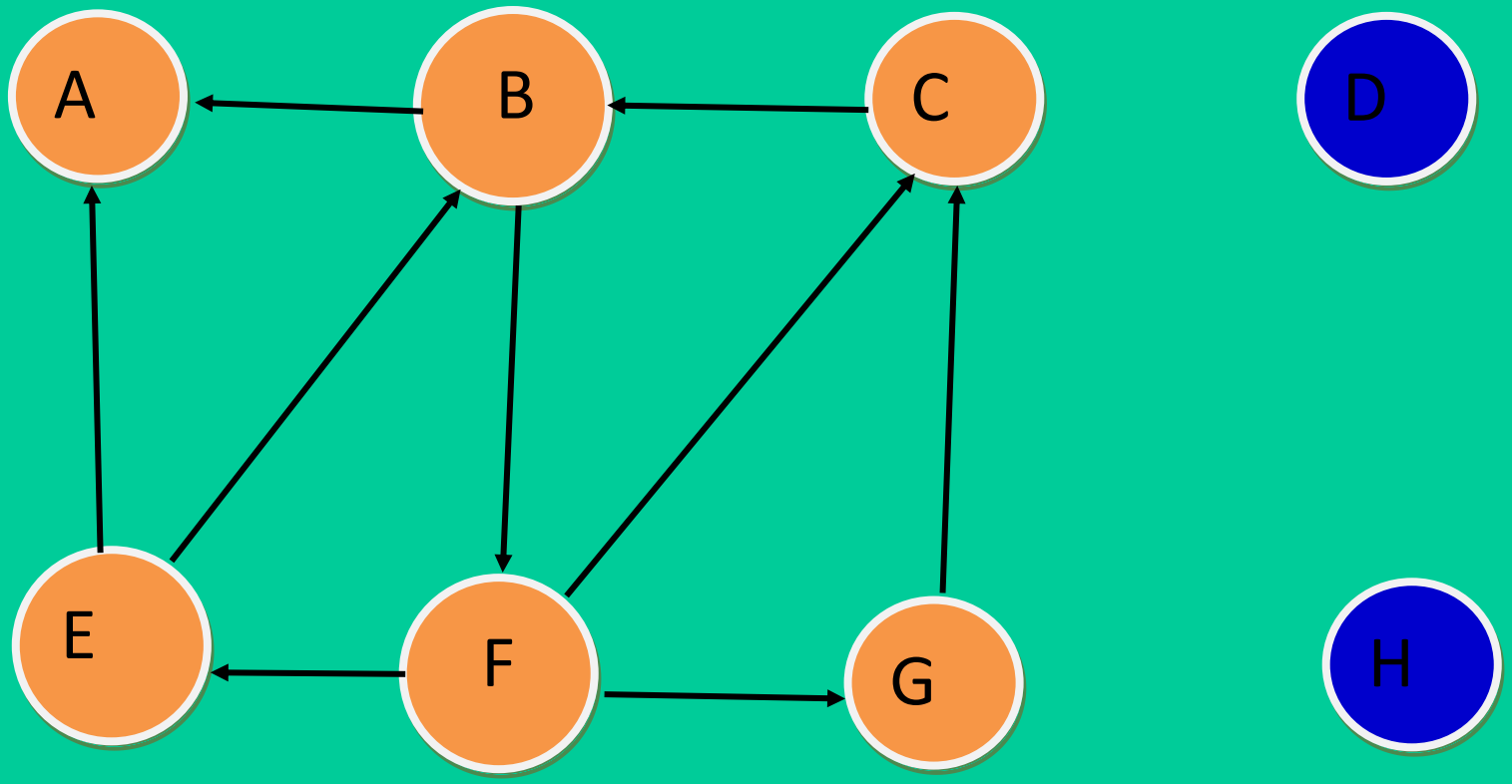
Lancer un parcours en profondeur en partant de $D_{(16)}$



D-H-A-B-E-C-G-F

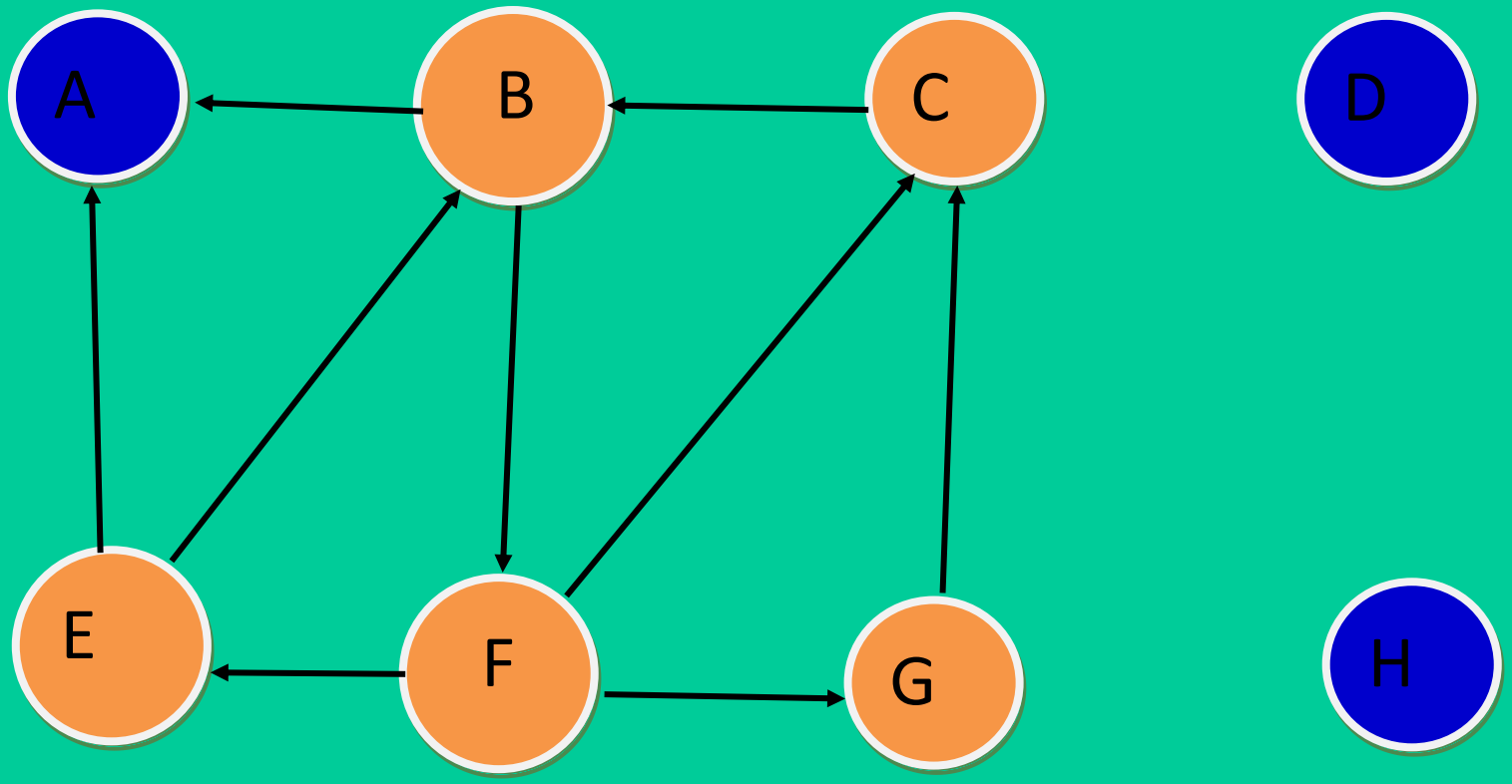

A
B
C
D
E
F
G
H

Lancer un parcours en profondeur en partant de $H_{(15)}$

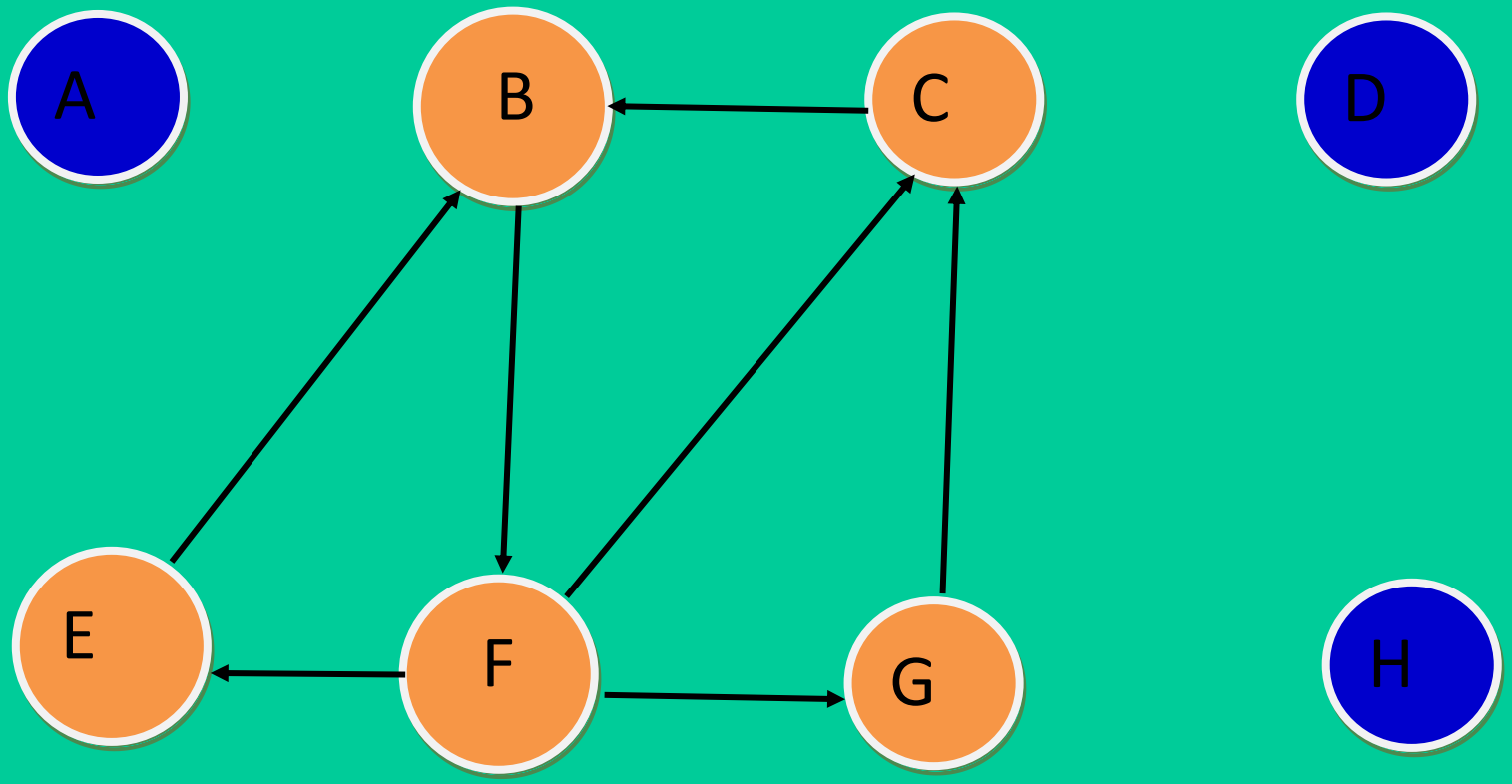


~~D~~-~~H~~-A-B-E-C-G-F

Lancer un parcours en profondeur en partant de **A**(12)

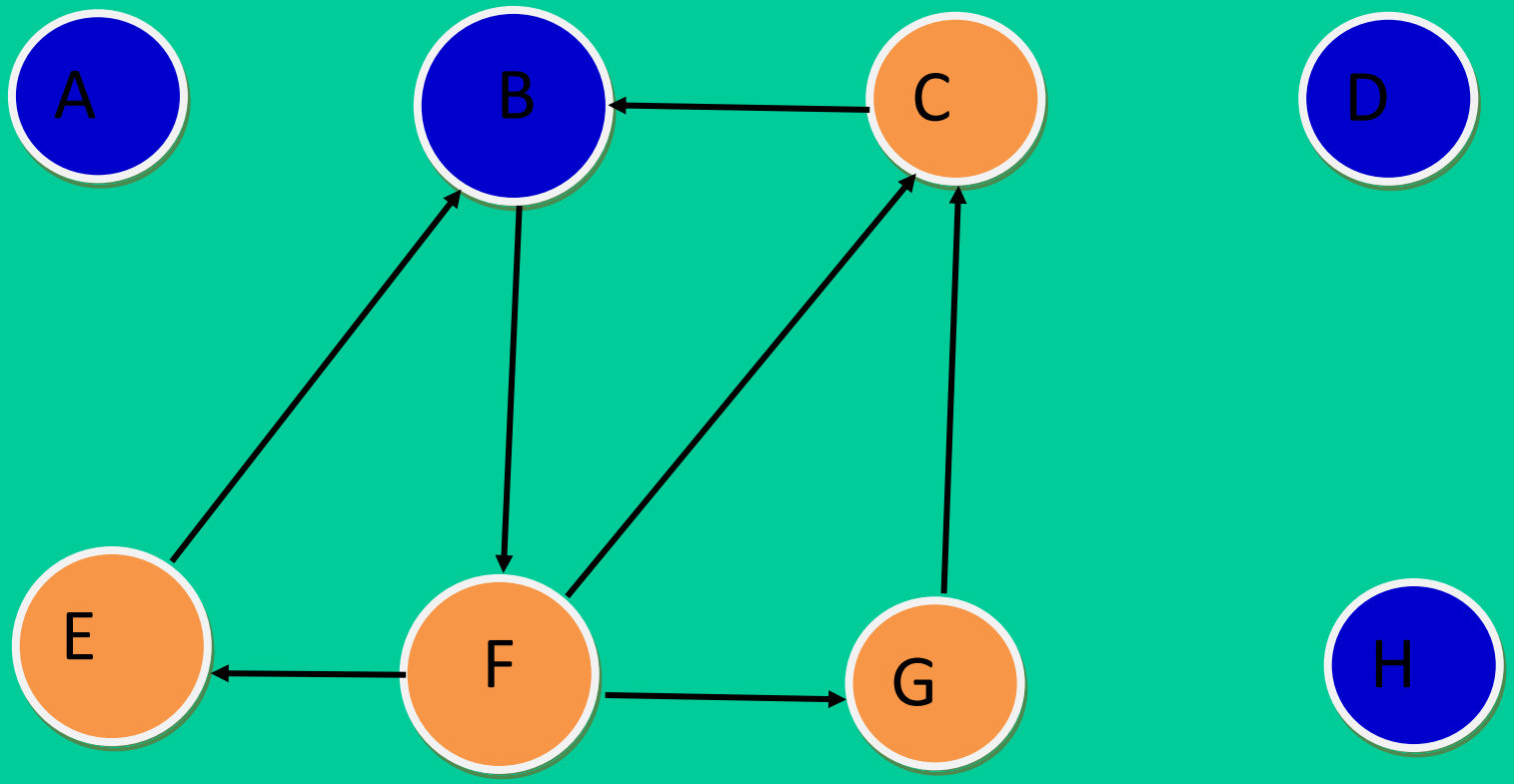


~~D~~-~~H~~-~~A~~-B-E-C-G-F

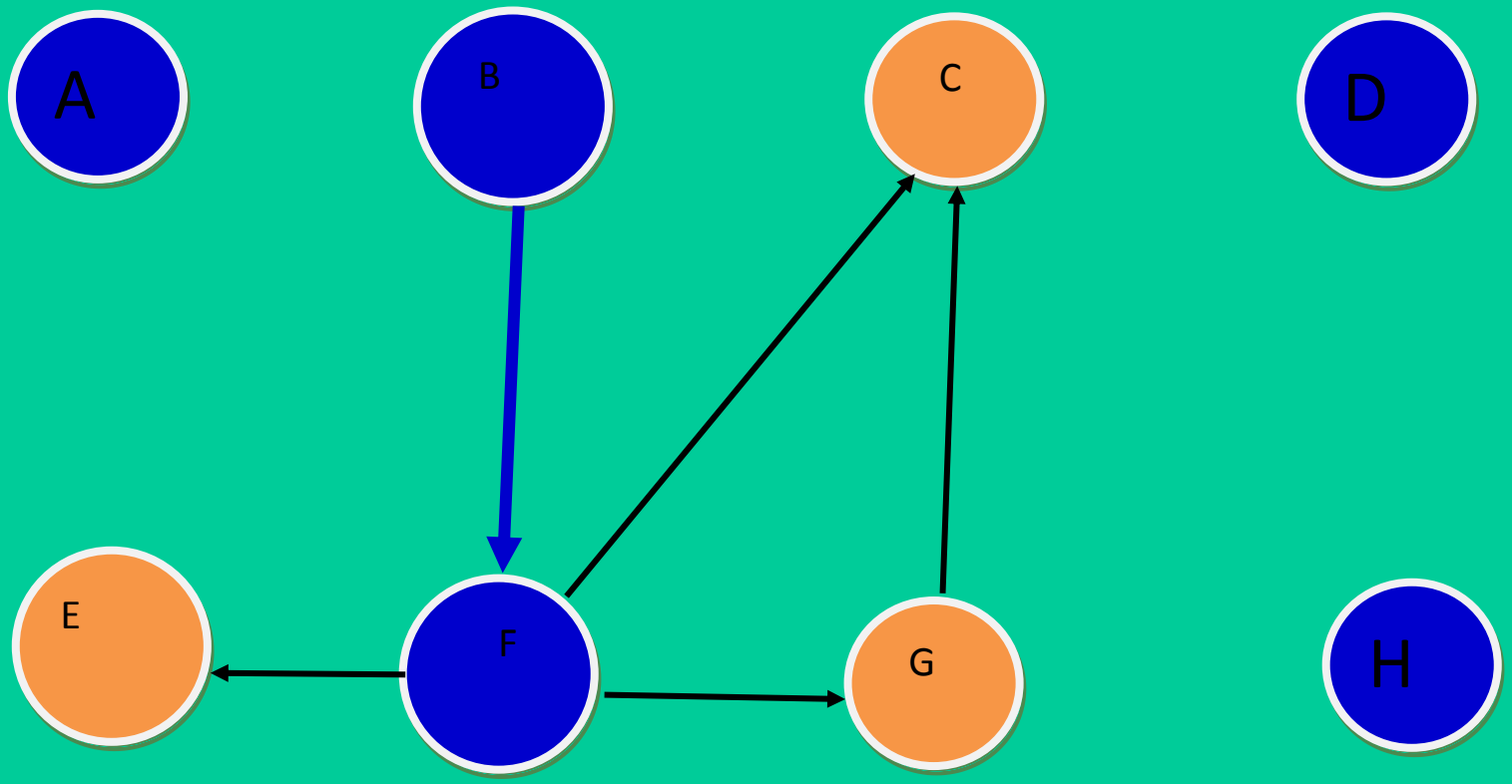


~~D~~-~~H~~-~~A~~-B-E-C-G-F

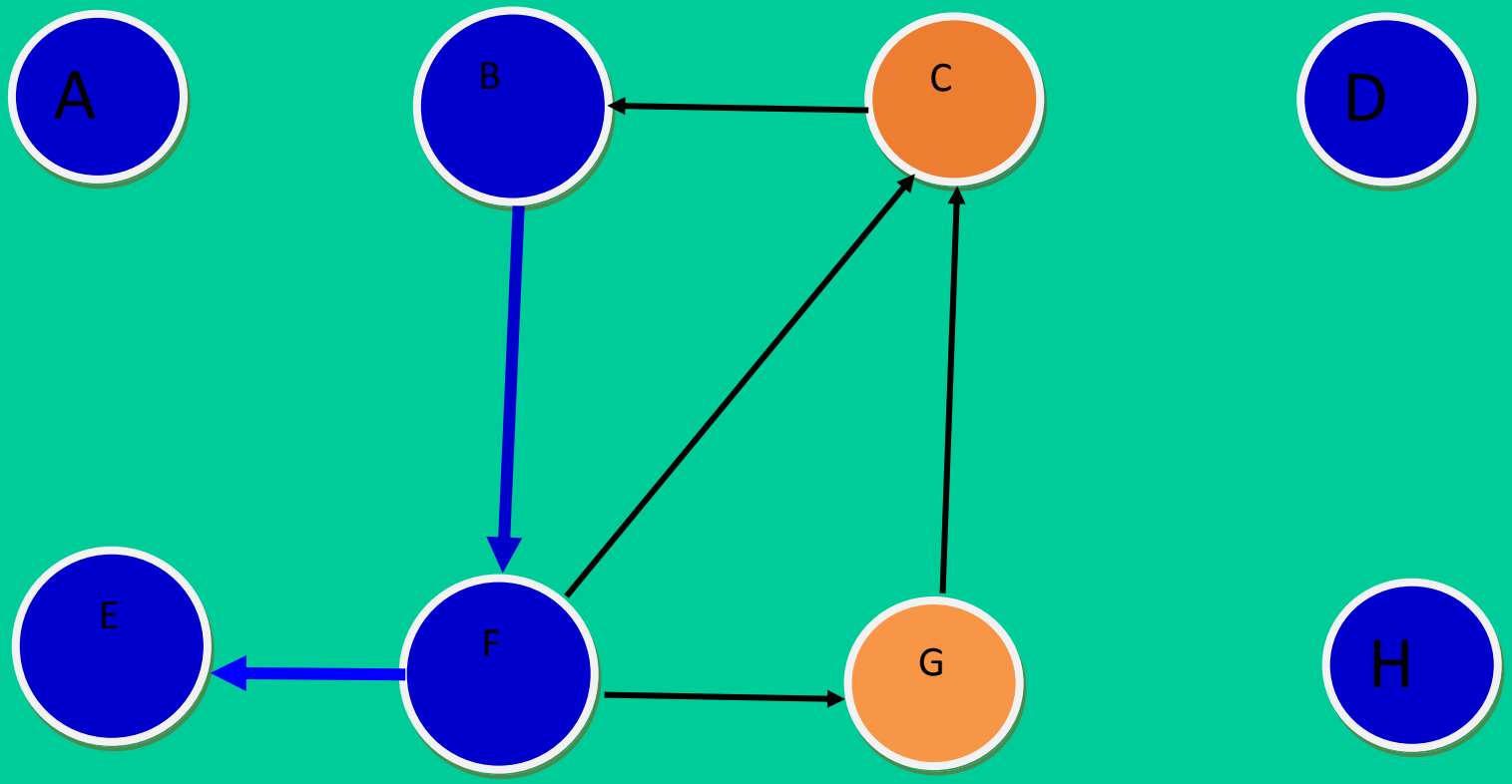
Lancer un parcours en profondeur en partant de $B_{(11)}$



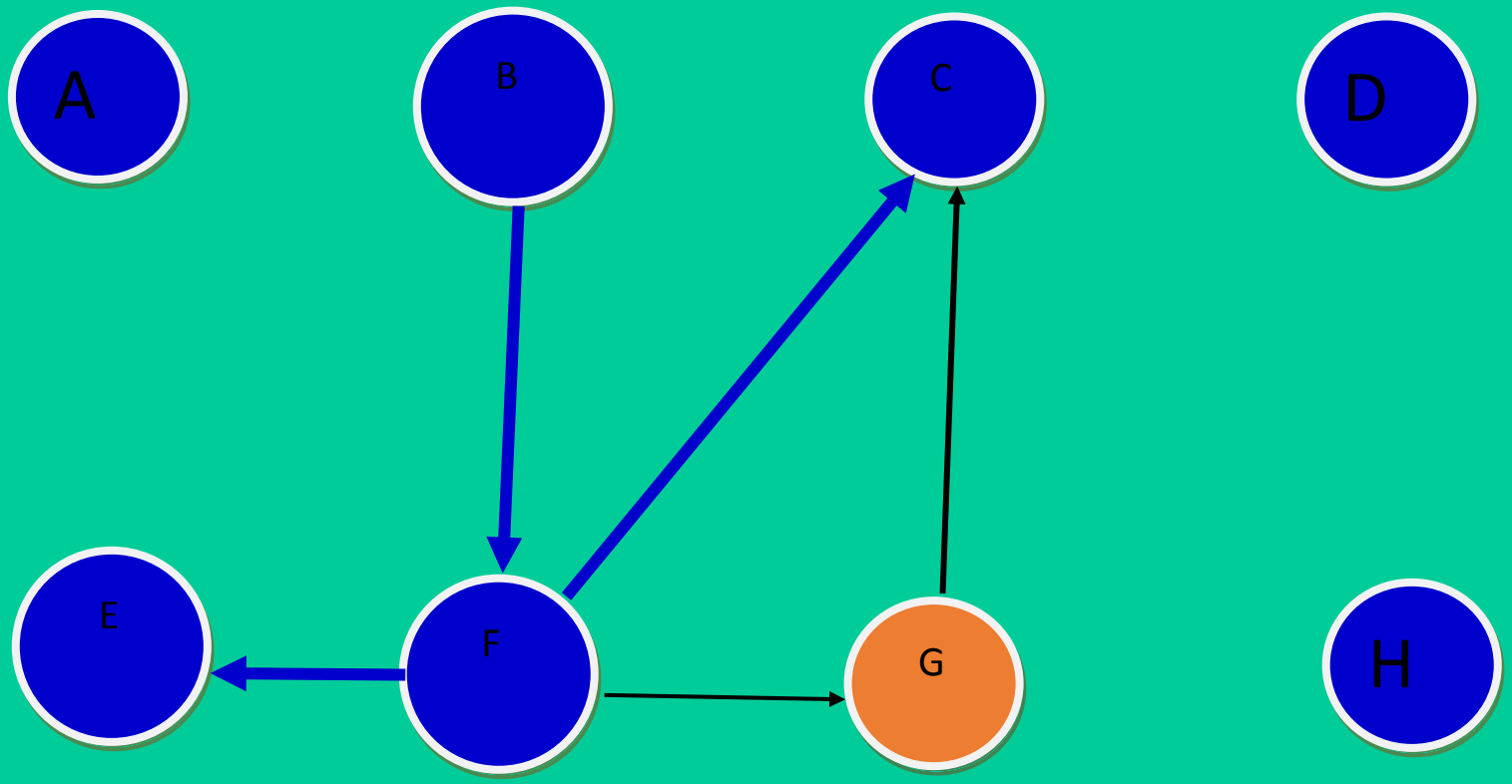
~~D~~-~~H~~-~~A~~-~~B~~-E-C-G-F



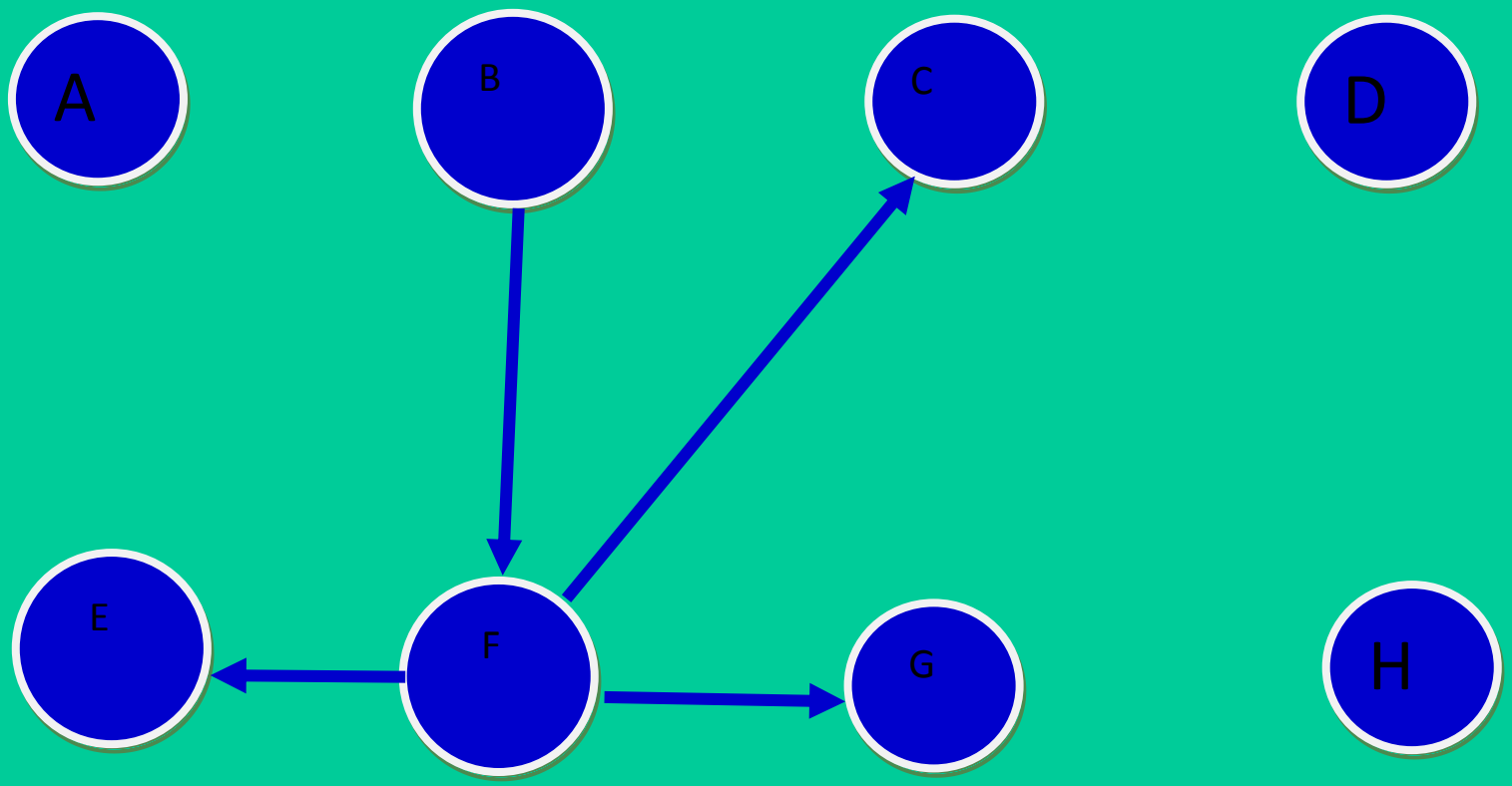
~~D-H-A-B~~-E-C-G-~~F~~



~~D-H-A-B-E~~-C-G-~~F~~

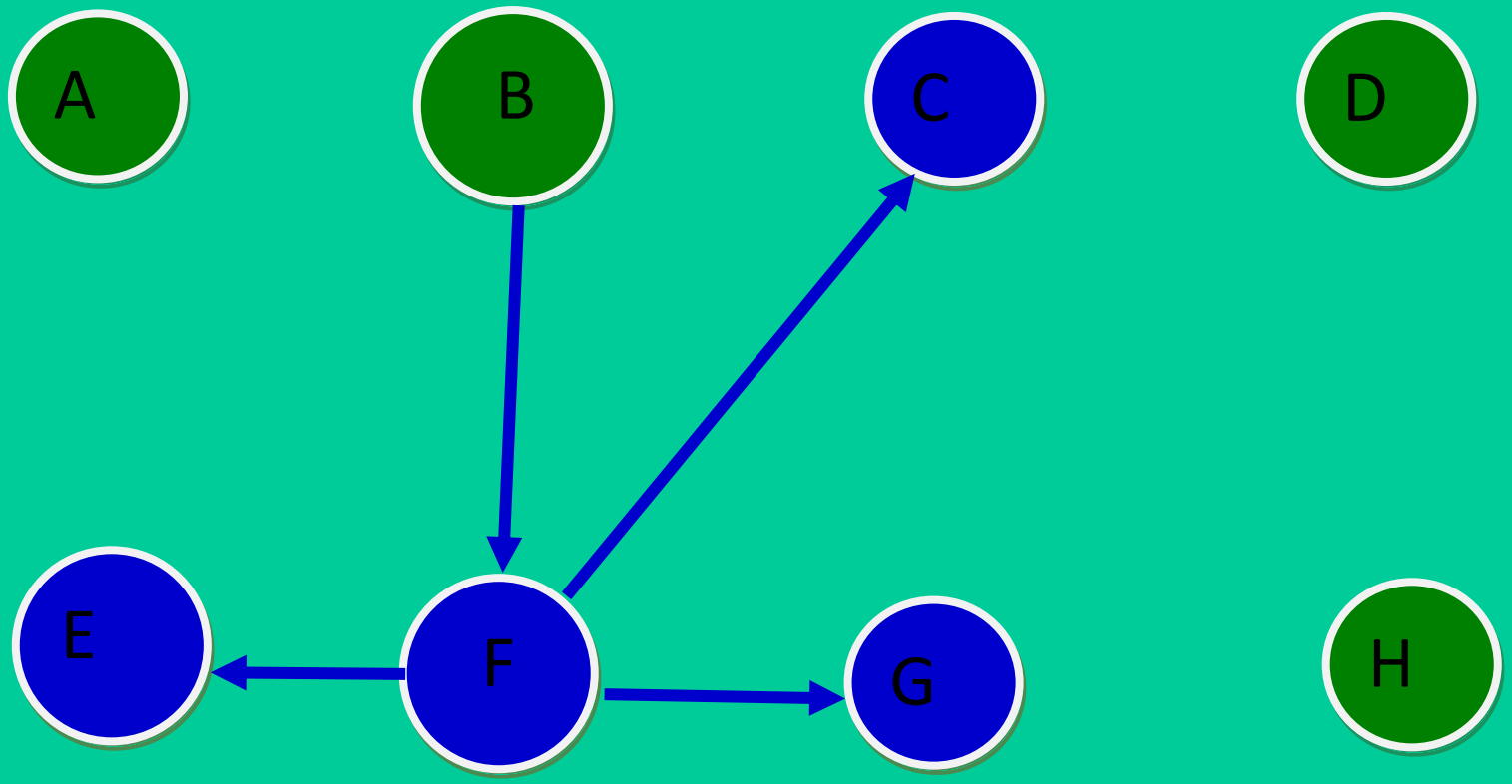


~~D-H-A-B-E-C~~-G-~~F~~

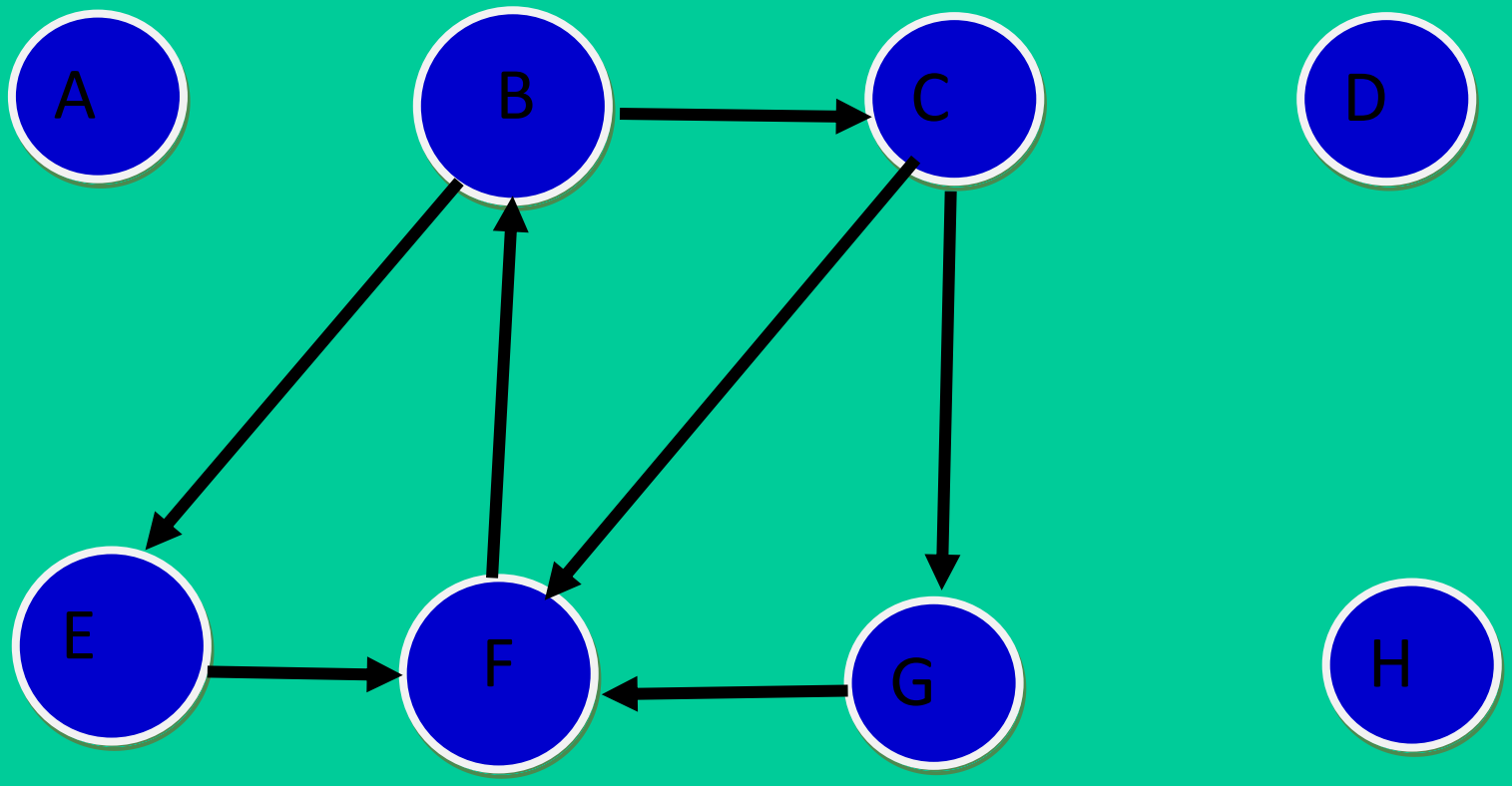


~~D-H-A-B-E-C-G-F~~

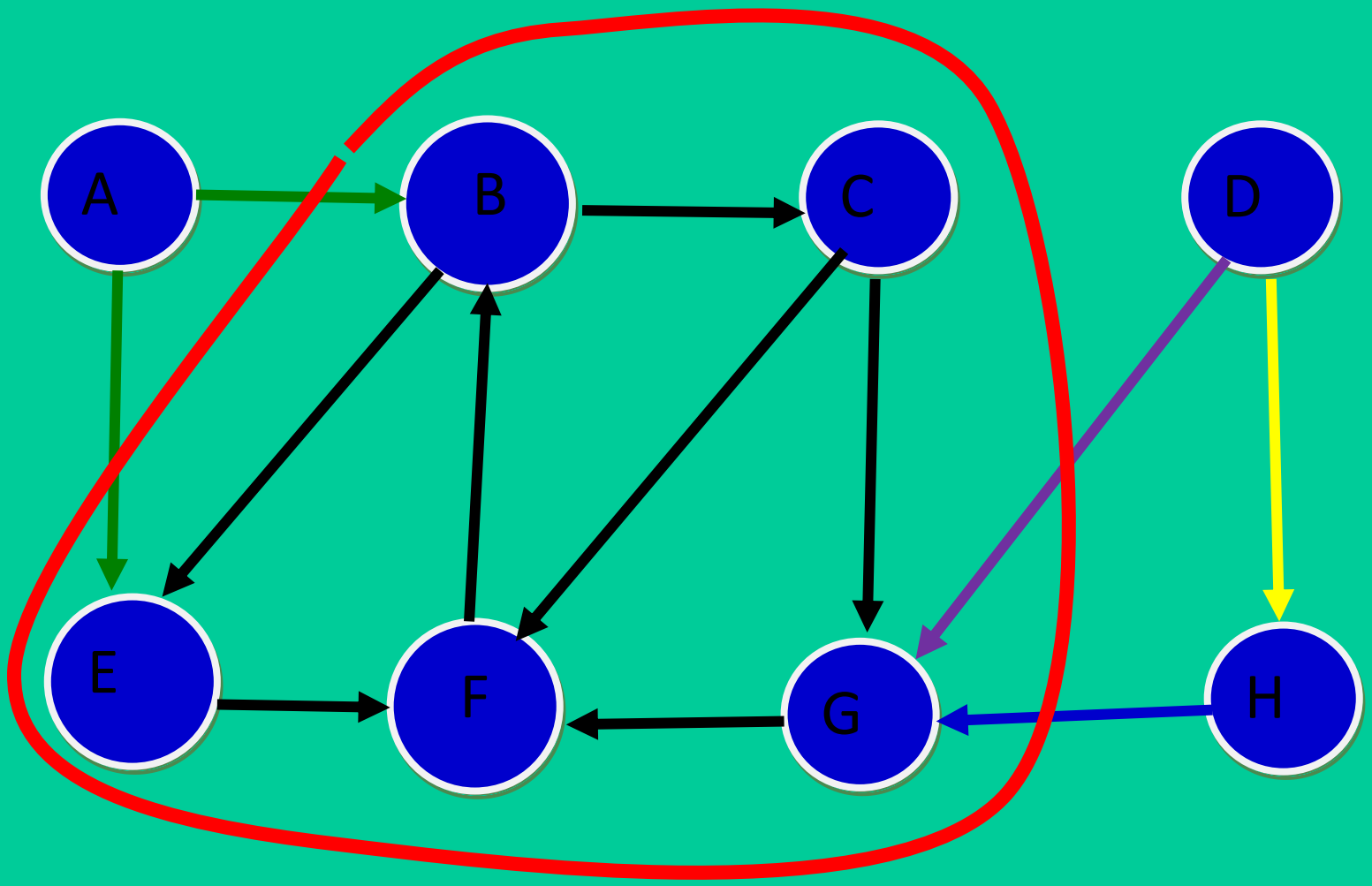
Le résultat est une forêt de 4 arborescences de racines A,B,D et H

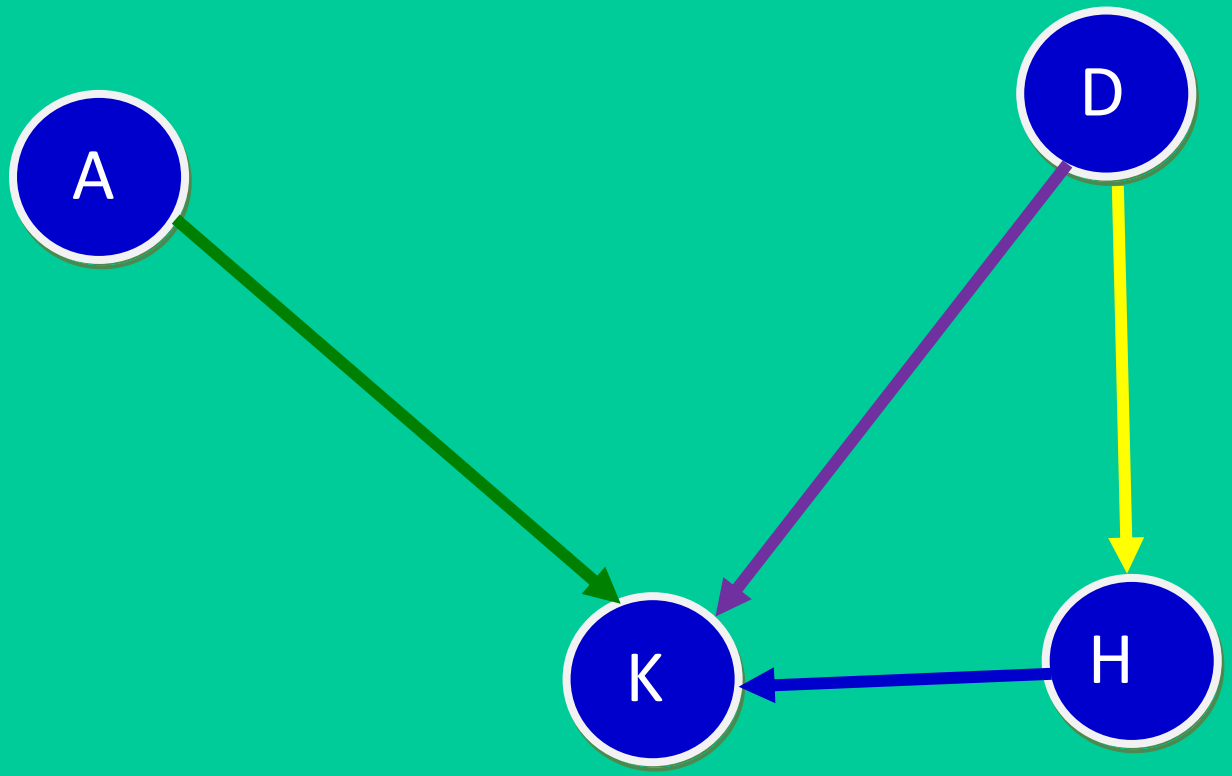


Leurs sommets sont ceux de 4 **composantes fortement connexes** dans le graphe G



Graphe quotient


A
B
C
D
E
F
G
H



K est un puits