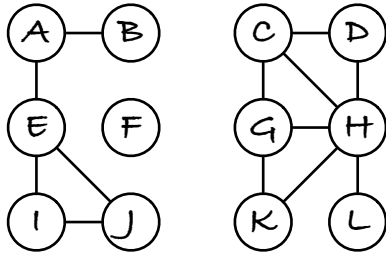
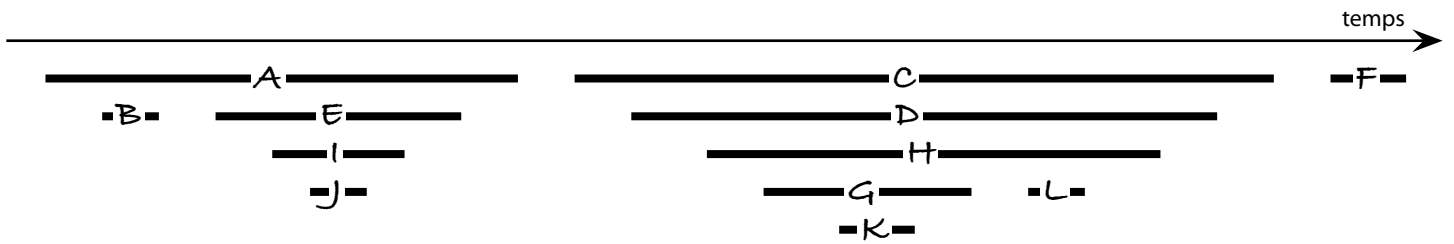
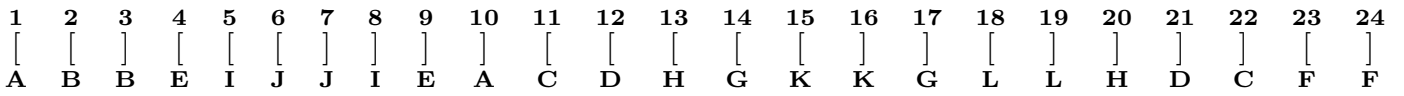
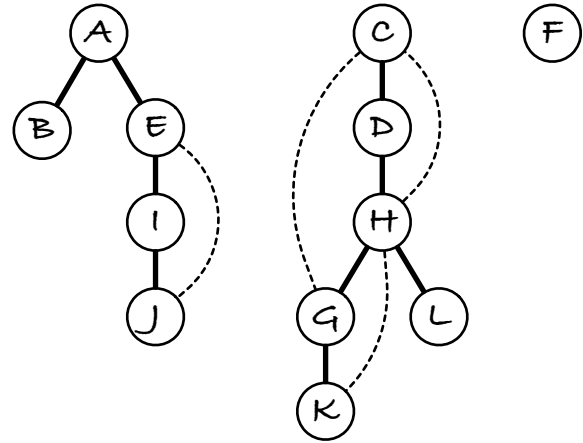


PARCOURS DE GRAPHES EN PROFONDEUR

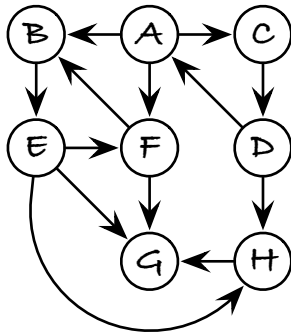
Un graphe non-orienté



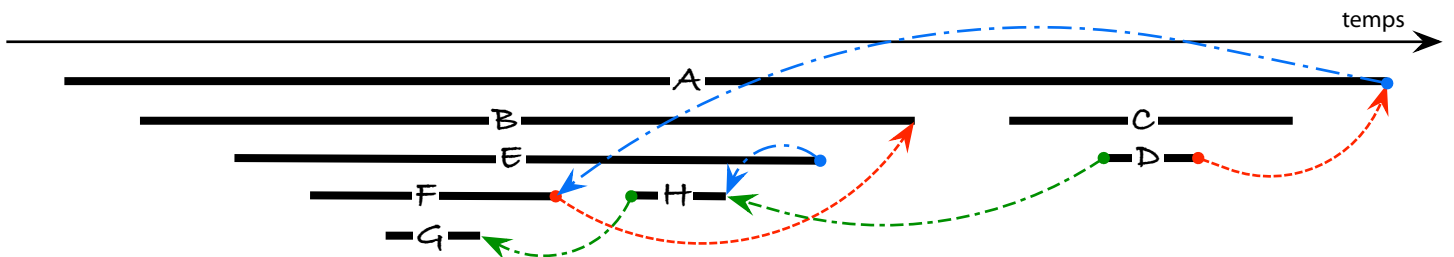
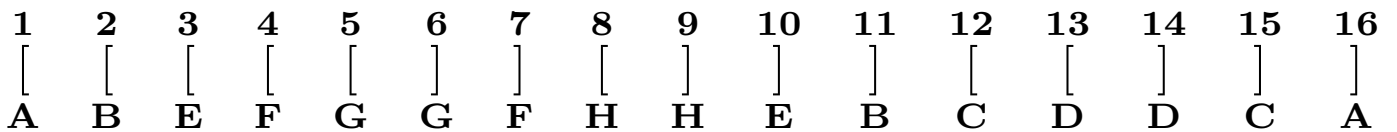
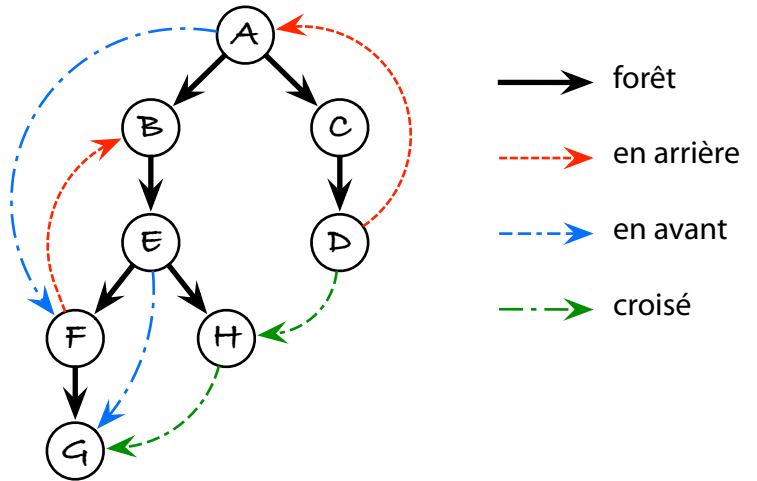
Sa forêt DFS issue d'un parcours en profondeur où les sommets sont parcourus dans l'ordre alphabétique



Un graphe orienté

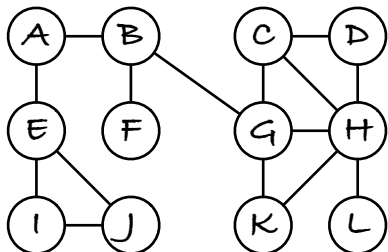


Sa forêt DFS issue d'un parcours en profondeur où les sommets sont parcourus dans l'ordre alphabétique

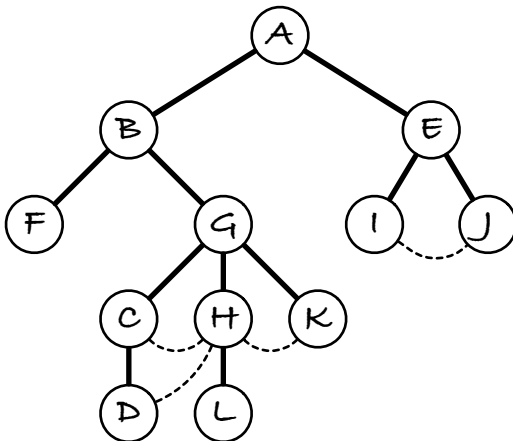


PARCOURS DE GRAPHES EN LARGEUR

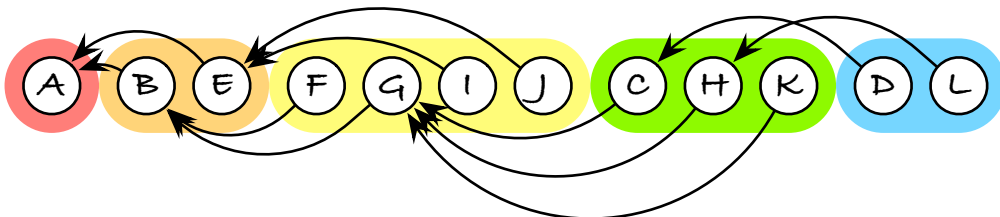
Un graphe non-orienté



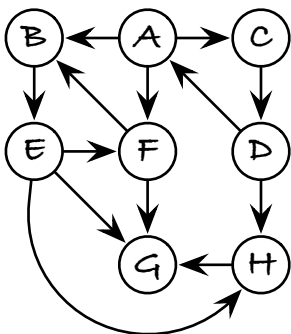
Sa forêt DFS issue d'un parcours en largeur
où les sommets sont parcourus dans l'ordre alphabétique



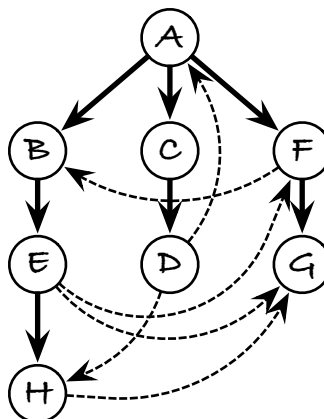
Contenu total de la file :



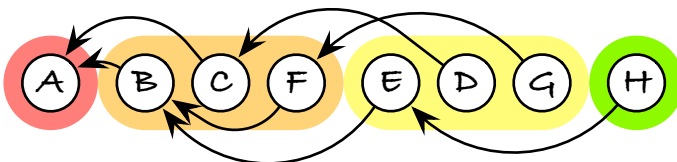
Un graphe orienté



Sa forêt DFS issue d'un parcours en largeur
où les sommets sont parcourus dans l'ordre alphabétique



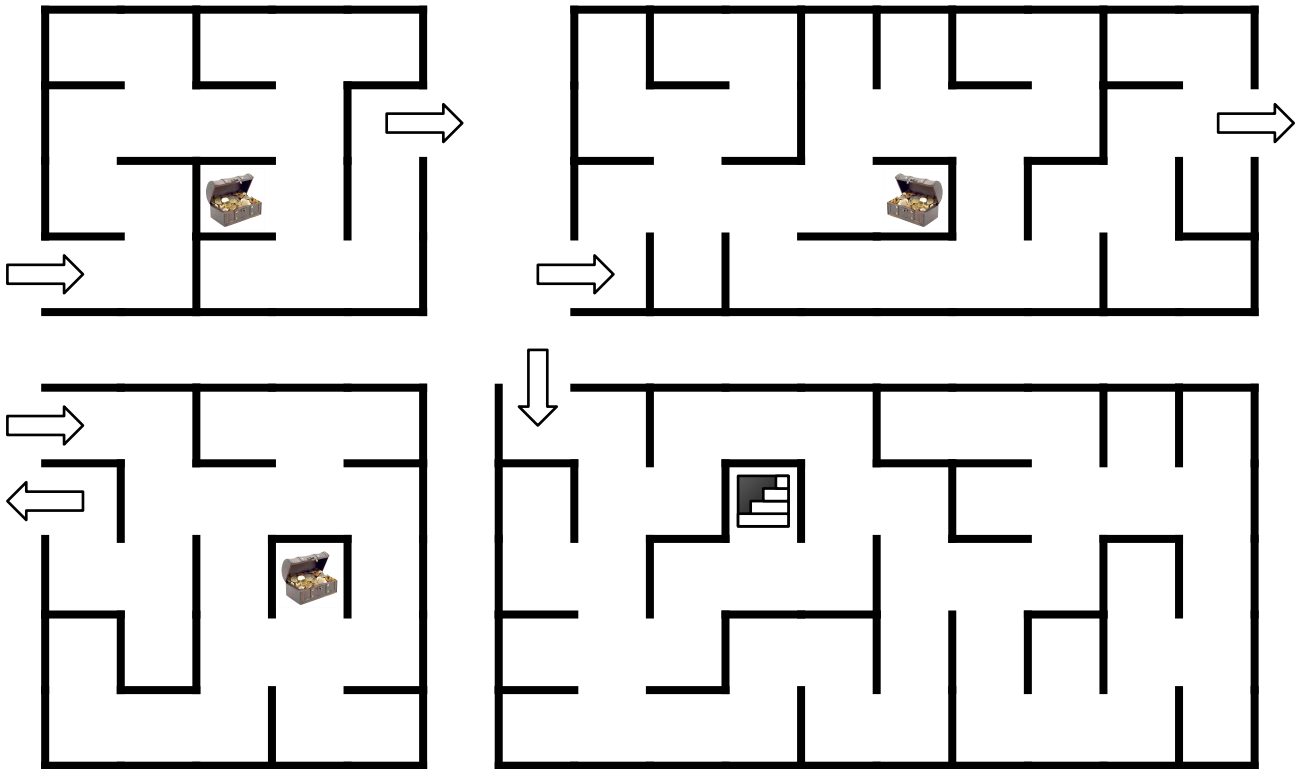
Contenu total de la file :



PARCOURS DE LABYRINTHES

Appliquez l'algorithme suivant :

Longer le mur de gauche jusqu'à trouver la sortie (c.-à-d. être sur la case de la sortie)



- L'algorithme précédent trouve-t-il toujours la sortie ? Proposez une condition sur la position de la sortie du labyrinthe pour que cet algorithme la trouve toujours.
- L'algorithme précédent passe-t-il toujours par la case du trésor ? Pouvez-vous caractériser la partie du labyrinthe que l'algorithme n'explore pas ? Proposez une condition sur le labyrinthe qui garantisse de trouver le trésor avec cet algorithme.
- Proposez un algorithme pour fabriquer un labyrinthe sans cycle. Exécutez l'algorithme ci-dessus sur un exemple de votre construction.
- Proposez d'utiliser des galets pour modifier l'algorithme pour qu'il trouve toujours la sortie et tous les trésors. Exécutez-le sur un exemple de votre construction.

