

TD - À la recherche de cycles - Graphes

Introduction

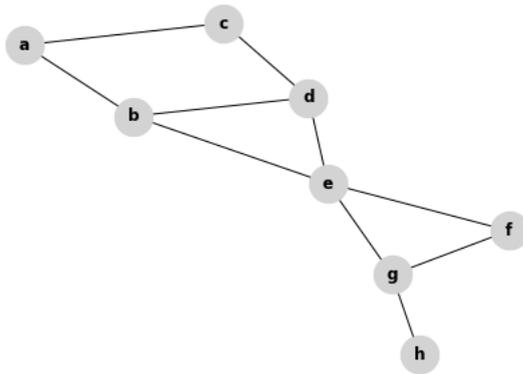
Rappels :

Une chaîne est une suite d'arêtes consécutives dans un graphe, un peu comme si on se promenait sur le graphe.

On la désigne par les lettres des sommets qu'elle comporte.

On utilise le terme de chaîne pour les graphes non orientés et le terme de chemin pour les graphes orientés.

Un cycle est une chaîne qui commence et se termine au même sommet.



a - b - e - d - c - a est un cycle

L'existence de cycle dans un graphe permet d'effectuer un trajet en revenant à son point de départ sans faire demi-tour...

REMARQUE :

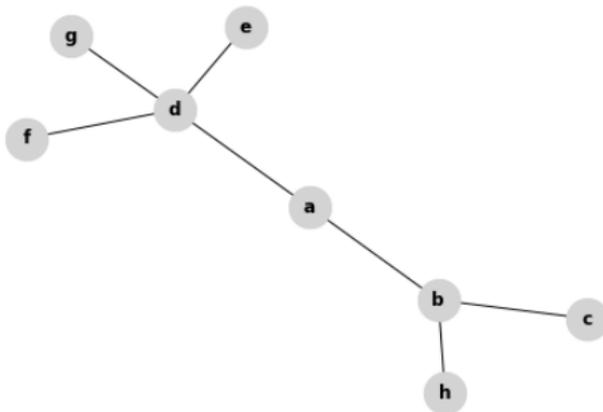
Pour un graphe connexe non orienté d'ordre n

Si le nombre d'arêtes est supérieur à $n - 1$, le graphe possède au moins un cycle.

À FAIRE 1:

Écrire une fonction qui détermine si un graphe connexe non orienté admet au moins un cycle.

Vérifier votre fonction pour le graphe précédent et celui-ci:



Détection de l'existence de cycles

Voici un programme qui détecte la présence d'un cycle dans un graphe connexe non orienté:

```
def detection_cycle(G, sommet):
    sommets_visites=[]
    f=File()
    sommets_visites.append(sommet)
    f.enfiler((sommet,-1))
    while f.vide() == False:
        (tmp,parent)=f.defiler()
        voisins=voisin(G,tmp)
        for vois in voisins:
            if vois not in sommets_visites:
                sommets_visites.append(vois)
                f.enfiler((vois,tmp))
            elif vois!=parent:
                return True
    return False
```

? QUESTION 1:

Quel type de parcours utilise ce programme?

.....

? QUESTION 2:

Quel est le type des éléments placés dans la file?

.....

✍ À FAIRE 2:

Faire fonctionner ce programme à la main sur ce graphe en partant du sommet a :

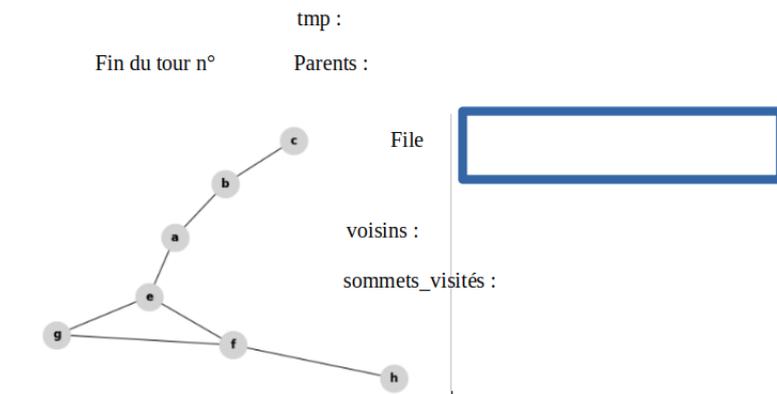
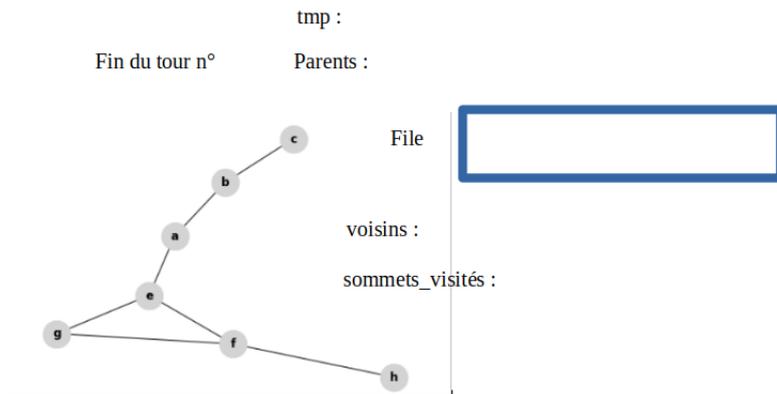
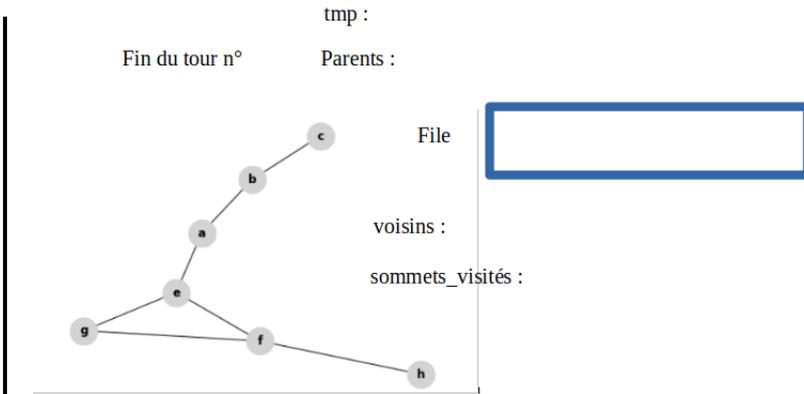
tmp :

Parents :

File:

voisins :

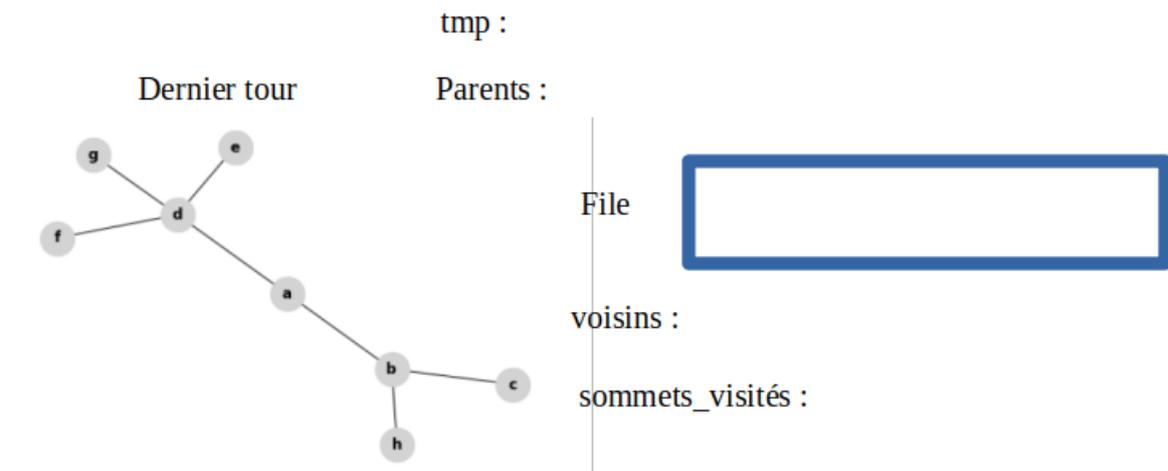
sommets_visités :



À FAIRE 3:

Pour finir de vous convaincre...

Faire fonctionner ce programme à la main pour le graphe ci-dessous en partant du sommet a. Noter les contenus des variables au dernier tour.



vérifications...

Vous pouvez vérifier vos résultats de "à la main" en ajoutant des `print` là où il faut dans le programme.

Pour faire afficher le contenu de la file : `print`(f.L)