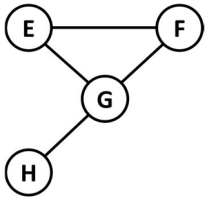
Cycle dans un graphe

N.S.I.

1. Définition



Une **chaine** (ou chemin) est une suite d'arêtes consécutives dans un graphe.
Un **cycle** est une chaine qui commence et se termine au même sommet (sans passer par la même arête).

- **E, F, G** est un cycle.
- **H, G, F, E** est une chaine.

Pour différentes raisons, il peut être intéressant de détecter la présence d'un ou plusieurs cycles dans un graphe (par exemple pour savoir s'il est possible d'effectuer un parcours qui revient à son point de départ sans être obligé de faire demi-tour).

2. Détection d'un cycle

L'algorithme de détection d'un cycle dans un graphe à partir d'un sommet.

```
On crée une liste vide de sommets visités
Dans la pile, on empile le sommet
Tant que la pile n'est pas vide
     On dépile dans la variable t
     Pour chaque voisin de t
          Si le voisin n'a pas été visité alors
               On l'empile
     Si t a été visité alors
          On retourne vrai
     Sinon on l'ajoute aux sommets visités
On retourne faux
```

✍ Implémenter la fonction **cycle(graphe, sommet)** correspondant à l'algorithme ci-dessus. Rajouter tous les éléments nécessaires à votre script pour pouvoir valider la fonction.

✍ Quel parcours (largeur ou profondeur) utilise cette fonction ?

✍ Créer un programme **detection_cycle(graphe)** qui utilise la fonction précédente, qui prend en paramètre un graphe et qui teste la présence ou non d'un cycle.

✍ Vérifier votre programme avec les deux graphes ci-dessous.

Absence de cycle

```
a = {}
a['C'] = ['D','E']
a['D'] = ['C']
a['E'] = ['C']
a['F'] = ['B']
a['A'] = ['B','C']
a['B'] = ['F']
```

Présence de cycle

```
g = dict()
g['A'] = ['B','C']
g['B'] = ['A','D','E']
g['C'] = ['A','D']
g['D'] = ['B','C','E']
g['E'] = ['B','D','F','G']
g['F'] = ['E','G']
g['G'] = ['E','F','H']
g['H'] = ['G']
```

Source : https://pixees.fr/informatiquelycee/n_site/nsi_term_algo_graphe.html