

Détection de cycle par récursivité

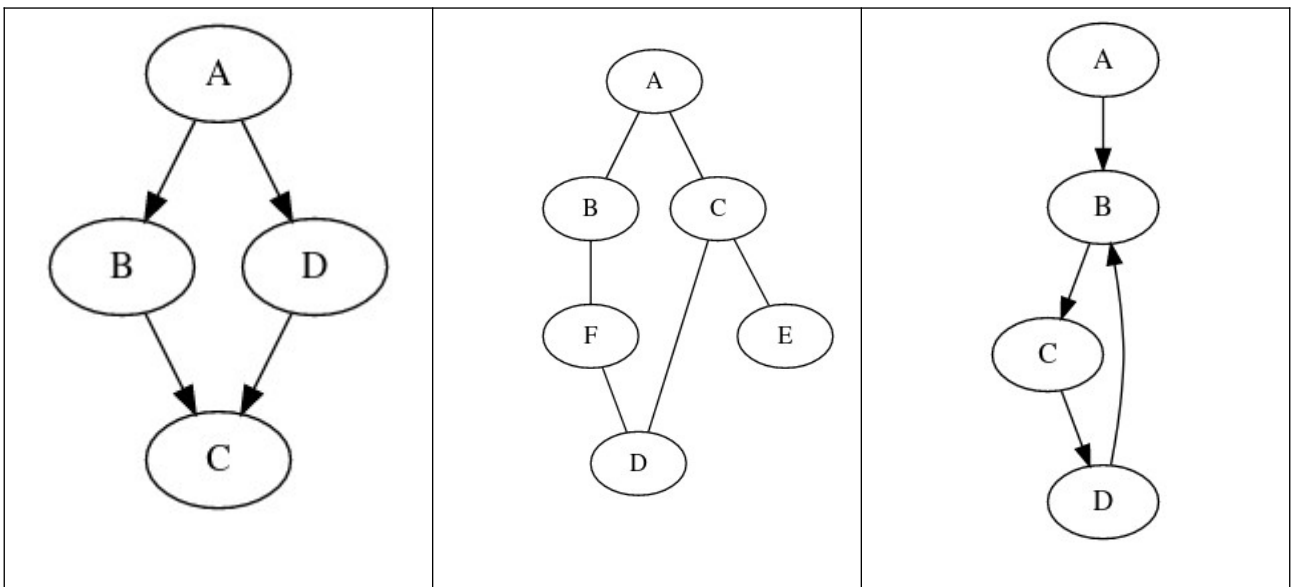
Algorithme :

Nous adopterons une solution en utilisant **3 couleurs** pour marquer ces sommets : blanc, gris et noir. Initialement, tous les sommets seront de **couleur blanche**.

Lorsqu'on visitera un sommet S :

- s'il est blanc :
 1. on colore le sommet S en gris ;
 2. on visite tous les voisins de S récursivement ;
 3. on colore le sommet S en noir une fois toutes les étapes précédentes franchies.
- s'il est gris, c'est qu'on vient de découvrir un cycle ;
- s'il est noir, on ne fait rien

Pour chacun des graphes ci-dessous, appliquez l'algorithme de détection d'un cycle au graphe ci-dessous (vous partirez du sommet de votre choix).



Pour chacun des graphes ci-dessus, écrire le dictionnaire des successeurs.

Implémentation :

BLANC, GRIS, NOIR = 1, 2, 3

#Remarque : Un dictionnaire est mutable, donc on peut le modifier par appels récursifs.

```
def parcours_cycle(graphe , couleur , s ) :
    # param : graphe (dict) dictionnaire des sucesseurs
    # param : couleur (dict) donne la couleur de chauqe sommet du graphe
    # param : un sommet de départ
    # return (bool)
    if couleur[s] == ... :
        return ....
    if couleur[s] == ... :
        return ....
    couleur[s] = ...
    for v in ....
        got_cycle = .....
        if got_cycle :
            return True
    couleur[s] = ...
    return False
```

```
def cycle(graphe) :
    couleur = {}
    for s in ... :
        couleur[s] = .....
    for s in ... :
        if ..... :
            return .....
    return .....
```

Tester l'implémentation sur les 3 graphes précédents.