Exercices parcours de graphe

Exercice 1 :

On dispose du graphe G ci dessous (à dessiner):
0: 2, 1, 3, 5
1: 3, 5, 4, 0, 2
2: 3, 0, 4, 1, 5
3: 1, 2, 5, 4, 0
4: 2, 5, 1, 3
5: 3, 1, 4, 2, 0

1. Dresser la liste d'adjacence du graphe G.
2. Rappeler l'algorithme de parcours en largeur d'abord.
3. Rappeler l'algorithme de parcours en largeur d'abord.
4. Réaliser un parcours en profondeur sur le graphe G en prenant le sommet 1 comme sommet de départ.
5. Réaliser un parcours en largeur sur le graphe G en prenant le sommet 0 comme sommet de départ.

Exercice 2 :

On dispose du graphe G ci dessous (à dessiner):
0: 1, 4
1: 3, 4, 2
2: 3, 0
3: 0, 4
4: 3, 2, 0, 1

1. Dresser la liste d'adjacence du graphe G.
2. Réaliser un parcours en profondeur sur le graphe G en prenant le sommet 3 comme sommet de départ.
3. Réaliser un parcours en largeur sur le graphe G en prenant le sommet 2 comme sommet de départ.

Exercice 3 :

On dispose du graphe G ci dessous.

0: 2, 4
1: 4, 3, 2
2: 4, 0, 3, 1
3: 1, 2, 4
4: 1, 2, 0, 3

1. Grâce à la classe Graphe Non-Orienté créée précédemment, créer le graphe non orienté correspondant.
2. Écrire une méthode *parcours_largeur* de la méthode Graphe Non Orienté qui implémante l'algorithme en langage naturel ci-dessous.
3. Écrire une méthode *parcours_profondeur* de la méthode Graphe Non Orienté qui implémante l'algorithme en langage naturel ci-dessous.

```
def parcours_profondeur(self, noeud_depart):
    deja_visites <- liste
    a_visiter = Pile
    a_visiter <- ajouter l'élément départ à la pile

    tant que la pile n'est pas vide et tous les sommets ne sont pas visités:
        sommet <- dépiler la pile
        afficher le sommet
        pour chaque voisin du sommet:
            si le voisin n'est pas déjà visité:
                a_visiter <- ajouter le voisin à la pile
```

```
def parcours_largeur(self, noeud_depart):
    deja_visites <- liste
    a_visiter = File
    a_visiter <- ajouter l'élément départ à la pile

    tant que la pile n'est pas vide et tous les sommets ne sont pas visités:
        sommet <- dépiler la file
        afficher le sommet
        pour chaque voisin du sommet:
            si le voisin n'est pas déjà visité:
                a_visiter <- ajouter le voisin à la file
```

Exercice 4

On dispose du graphe orienté G ci-dessous:

0 : 1,2
1 : 3,4
2 : 3
3 : 4
4 : ∅

1. Grâce à la classe Graphe Orienté créée précédemment, créer le graphe orienté correspondant.
2. Écrire une méthode *parcours_largeur* de la méthode Graphe Orienté.
3. Écrire une méthode *parcours_profondeur* de la méthode Graphe Orienté.