Université Alger 1 Benyoucef Benkhedda

Faculté des sciences

Département informatique



**Analyse des réseaux sociaux (M2 ASD)**

**TP1**

**Graphes sous Python**

**2023-2024**

Date de remise : 28 octobre 2023

|  |  |
| --- | --- |
| **Consignes** | |
| Bibliothèque python | Utiliser la bibliothèque NetworkX dans Python pour la réalisation de ce TP. |
| Remise | Date de remise à déterminer par la chargée de cours |
| Rendu | Code source, et résultat de l’exécution |
| Type de remise | Fichiers électroniques |

# Partie 1 : Base de la théorie des graphes

Le but de ce premier exercice est simplement de mettre en ÷uvre les diérentes notions introduites précédemment :

Créer un graphe G et un graphe orienté DG.

1. Ajouter à chaque graphe les sommets nommés 1, 2, 3, 4 et 5.
2. Afficher les sommets des graphes.
3. Supprimer le sommet 1 du graphe G.
4. Ajouter au graphe G les arêtes {2, 3}, {2, 5}, {3, 4} et {4, 5}.
5. Ajouter au graphe DG les arêtes (1, 3), (2, 3), (2, 4), (2, 5), (4, 5) et (5, 1).
6. Tracer le graphe G et le graphe DG.
7. Calculer et afficher la taille de G
8. Calculer et afficher l’ordre de G et de DG
9. Déterminer et afficher la matrice d’adjacence de G
10. Pondérer les arêtes de DG, d’une façon aléatoire avec des valeurs comprise entre 1 et 4
11. Déterminer et afficher la matrice d’adjacence de DG
12. Afficher les listes d’adjacence de DG

# Partie 2 : Parcours en profondeur

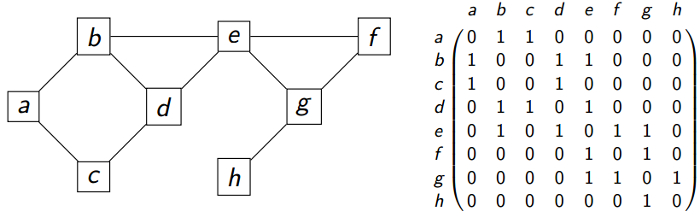
Le parcours en profondeur d’un graphe à partir d’un sommet consiste à suivre les arêtes arbitrairement, en marquant les sommets déjà visités pour ne pas les visiter à nouveau.

**Algorithme**

Afin de mémoriser les sommets visités pendant le parcours, on les marque, en les plaçant dans une liste ou un ensemble par exemple.

* Initialement, on marque le sommet de départ comme « visité »
* On choisit ensuite arbitrairement une de ses arêtes sortantes
  + Si le sommet voisin n’a pas déjà été visité :
    - on le marque comme « visité »
    - on relance un nouveau parcours à partir de ce sommet
* on recommence avec son arête sortante suivante …
* le parcours s’arrête lorsqu’on n’atteint plus aucun sommet non visité.

Soit le graphe G suivant, avec sa matrice d’adjacence



1. Implémenter G en utilisant un dictionnaire
2. Écrire une fonction ParcourirProfondeur(Graphe,Sommet,Visitee), qui permet de parcours un graphe G à partir d’un nœud N.
3. Appliquer votre fonction sur le graphe G, en partant du nœud a, ensuite du nœud f.

# **Livrable :**

1. Code source sous Python
2. Copie de la console lors des exécutions du code
3. Remise dans Classroom