

Introduction

Comme nous l'avons vu dans les TPs précédents, un graphe (orienté ou non) peut être composé de noeuds (ou sommets) attribués, mais aussi d'arêtes (ou arcs) attribués (ou étiquetés).

La librairie `networkx` disponible sur Python permet de créer des graphes orientés ou non, attribués, pour les noeuds/sommets comme pour les arêtes/arcs.

Durant les TPs précédents, vous avez implémenté des méthodes pour extraire différentes caractéristiques d'un graphe mais aussi parcourir ce graphe. Ces méthodes sont pré-implémentées dans `networkx`. L'idée de ce TP est donc d'utiliser les méthodes de la librairie au lieu de les implémenter soi-même.

[Si vous vous demandez à quoi ont servi les précédents TPs puisque tous les algorithmes sont déjà pré-implémentés, c'est pour que vous soyez **légitimes dans vos compétences** (il faut savoir comprendre un algorithme avant de pouvoir l'appliquer).]

Afin de pouvoir profiter pleinement de cette librairie, pour la suite de ce TP, vous devrez installer la librairie sur votre ordinateur. Pour cela, dans le terminal de votre logiciel de programmation (VSCode, PyCharm, Spyder, ...), entrez une de ces deux lignes de commande suivantes :

```
1 pip install networkx # ou
2 python -m pip install networkx
```

Pour ceux qui utilisent Jupyter Notebook, vous pouvez entrer la ligne de commande suivante dans une cellule que vous aurez créée :

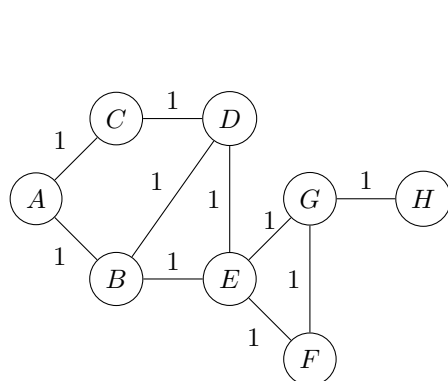
```
1 !pip install networkx
```

Si l'installation ne fonctionne pas de votre côté, n'hésitez pas à me demander de l'aide rapidement.

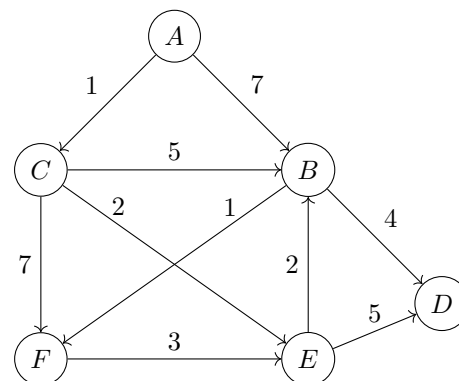
[Si vous utilisez Replit, vous n'avez pas besoin de lancer la ligne de commande précédente, il suffit de l'importer directement dans votre fichier `.py`, Replit se chargera de l'installer et de l'importer.]

Comme dans le TP1, vous devrez importer la librairie `networkx` de la même manière que la librairie `numpy` en début de code :

```
1 import networkx as nx
```



Graphe non orienté.



Graphe orienté.

Exercice 1

Construisez le graphe non orienté à l'aide de la librairie `networkx`, en ajoutant chaque sommet puis chaque arête un par un.

Indication : Vous aurez besoin d'instancier l'objet `Graph()` et entres autres, des méthodes `add_node` et `add_edge`, tous disponibles directement via `networkx`.

Reconstruisez le graphe non orienté en ajoutant les sommets en une seule action. Puis, faites de même pour les arêtes.

Dessiner le graphe.

Exercice 2

Affichez l'ensemble des sommets et des arêtes. Affichez le degré associé à chaque sommet. Affichez la matrice d'adjacence du graphe non orienté. Affichez les sommets adjacents de chacun de ces sommets : A , D , E et H .

Exercice 3

Supprimez les sommets B et G du graphe. Pour la suite du TP, assurez-vous de rajouter ces sommets à nouveau, ainsi que les arêtes associés pour obtenir le graphe initial.

Exercice 4

Vérifiez s'il existe un chemin entre ces couples de sommets : (C, F) , (A, G) et (B, H) .

Exercice 5

Affichez le plus court le chemin entre ces couples de sommets : (C, F) , (A, G) et (B, H) . Affichez les chemins et leur longueur.

Exercice 6

Refaites tous les exercices précédents, mais avec le graphe orienté.

Indication : L'objet `Graph()` instancie un graphe non orienté tandis que l'objet `DiGraph()` instancie un graphe orienté (en anglais *Directed Graph*). L'ajout de noeuds/sommets et d'arcs/arêtes se fait de la même manière que pour un graphe non orienté, il y a juste un sens à respecter pour les arcs/arêtes ($V_{src} \rightarrow V_{dst}$).

Pour l'exercice 2 avec le graphe orienté, affichez les sommets adjacents de chacun de ces sommets suivants : A , C , D et F .

Pour l'exercice 4 avec le graphe orienté, vérifiez l'existence d'un chemin pour ces couples de sommets suivants : (A, E) , (A, F) et (F, A) .

Pour l'exercice 5 avec le graphe orienté, affichez le plus court chemin pour ces couples de sommets suivants : (A, E) , (B, E) et (C, D) .

Exercice 7 (pour le *fun*)

Parcourez en profondeur le graphe orienté à partir du sommet A . Faites de même avec un parcours en largeur. Affichez les parcours (les arêtes et leur longueur associée).