SDD_Graphes

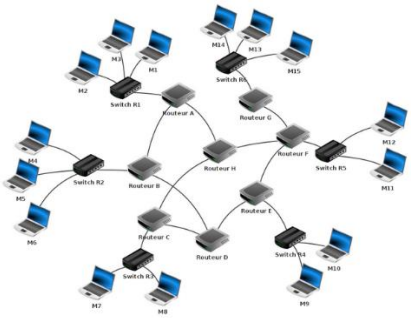
I/ Vocabulaire des graphes

1/ Exemples et définition

On appelle **graphe** la donnée d'un ensemble fini V de sommets et d'un ensemble E de liens entre eux.

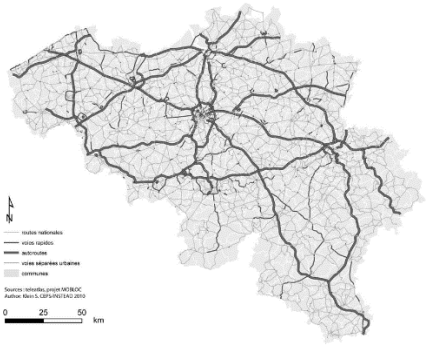
Quelques exemples de graphes au quotidien :

Réseau Internet



Réseau social

Réseau routier

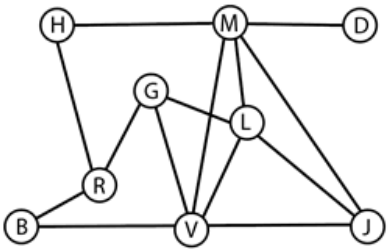


Les **liens** sont le choix d'un sommet de départ (appelé S1 ici) et d'un sommet d'arrivée (appelé S2 ici).

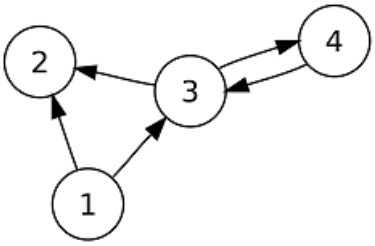
- Si ces liens sont symétriques, c'est-à-dire que l'on, peut aller de S1 à S2 mais aussi de S2 à S1, le graphe est **non-orienté** : chacun d'entre eux peut être représenté par une arête.
- A l'inverse, le graphe est **orienté** et les chemins possibles sont représentés par des arcs.
- Des poids peuvent être associés aux liens d'un graphe (orienté ou non). On parle alors de graphes **pondéré**.

Quelques exemples pour illustrer les définitions ci-dessus :

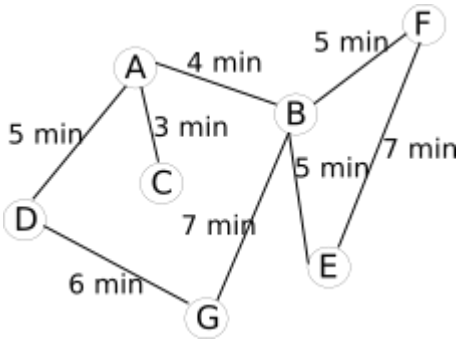
Graphe non orienté



Graphe orienté

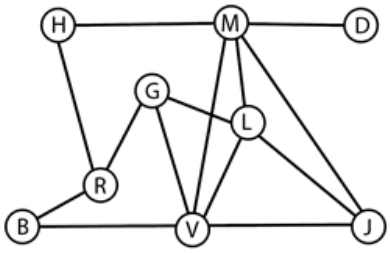


Graphe pondéré



2/ Graphes non orientés

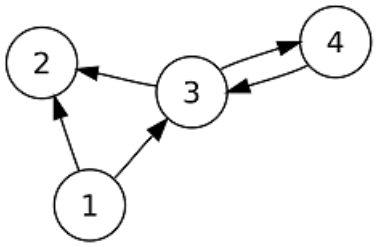
- Les liens d'un graphe non orienté s'appellent des **arêtes**.
- Une **chaîne** est une suite (finie) consécutives d'arêtes. Par exemple, D-M-J-V-G sur le graphe à droite.
- Un graphe est **connexe** s'il existe une chaîne reliant entre tous sommets.
- Si une **chaîne** mène d'un sommet à lui-même , on parle alors de **cycle**. Par exemple H-M-V-B-R-H sur le graphe à droite.



Remarque : un graphe ne comportant aucun cycle est un arbre.

3/ Graphes orientés

- Les liens d'un graphe orienté s'appellent des **arcs**.
- Un **chemin** est une suite (finie) consécutives d'arcs. Par exemple, 4-3-2 sur le graphe à droite.
- Un graphe est **connexe** s'il existe un chemin pour tous sommets, chaque arc étant transformé en arête. C'est le cas pour le graphe à droite.



II/ Implémentations des graphes

1/ Différentes méthodes

Plusieurs modes de représentation peuvent être implémentées pour stocker des graphes : matrices d'adjacences, liste des voisins, des successeurs ou des prédécesseurs. Une matrice notée

2/ Matrice d'adjacence

Rappel :

Une **matrice** est un tableau de nombres et peut être représentée par une liste de listes en langage Python.

La matrice A:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Représentation en Python

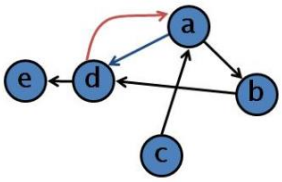
```
A = [[1,2,3],[4,5,6],[7,8,9]]

print(A[1][2])  # Attendu : 6
```

```
6
```

Un graphe peut être représenté par une **matrice d'adjacence** : pour **tout lien existant** entre deux sommets (ce sont des sommets dits adjacents ou voisins), on donne la valeur **« 1 » sinon** on donne **« 0 »**.
Si le graphe est pondéré, on peut indiquer la valeur de chaque poids dans la matrice.

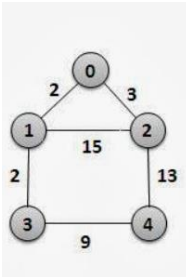
Voici un exemple de graphes avec leur matrice d'adjacence :

Graphe non pondéré



	a	b	c	d	e
a	0	1	0	1	0
b	0	0	0	1	0
c	1	0	0	0	0
d	1	0	0	0	1
e	0	0	0	0	0

Graphe pondéré



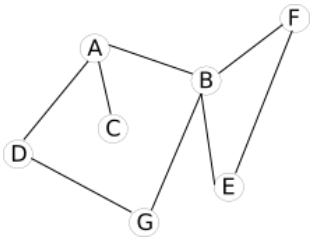
	0	1	2	3	4
0	0	2	3	0	0
1	2	0	15	2	0
2	3	15	0	0	13
3	0	2	0	0	9
4	0	0	13	9	0

3/ Liste d'adjacence

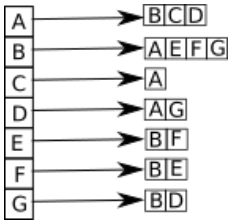
Pour commencer, on définit une liste des sommets du graphe. À chaque élément de cette liste, on associe une autre liste qui contient les sommets lié à cet élément.

Voici un exemple :

Un graphe

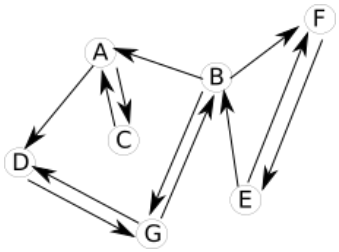


et sa liste d'adjacence (ou des voisins)

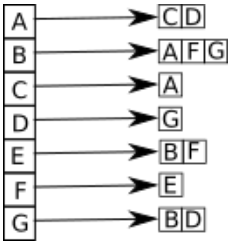


Pour les graphes **orientés**, il est nécessaire de définir 2 listes : la liste des successeurs et la liste des prédécesseurs. Soit un arc allant d'un sommet B vers un sommet A (flèche de B vers A). On dira que B est un successeur de A et que A est un prédécesseur de B.

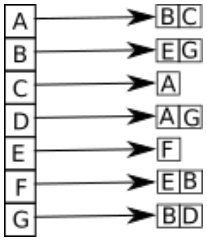
Un graphe orienté



Liste des successeurs



Liste des prédécesseurs

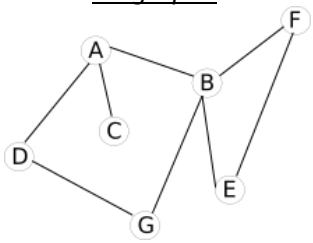


Source : site pixies de David Roche

Il est possible de travailler avec des listes d'adjacences en Python en utilisant les **dictionnaires** :

Voici un exemple :

Un graphe



Dictionnaire en Python

```
l = {'A':('B','C','D'), 'B':('A', 'E', 'F', 'G'), 'C':('A'), 'D':('A', 'G'), 'E':('B', 'F'), 'F':('B', 'E'), 'G':('B', 'D')}
```

Le choix entre matrice d'adjacence et liste des voisins se fait surtout en fonction de la **densité** du graphe (nombre important de liens) : plus il est dense, plus la matrice d'adjacence devient intéressante.
Le type d'algorithme utilisé influence également ce choix.