Graphes

Notions de base, modélisation d’un graphe

Anicet E. T. Ebou, ediman.ebou@inphb.ci

01

Introduction aux graphes

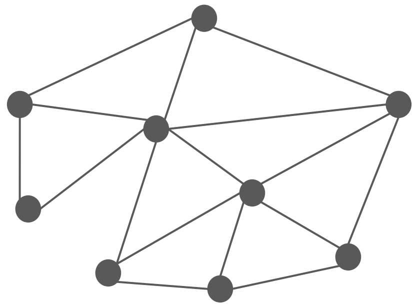
Qu’est-ce qu’un graphe ?

Un graphe est un ensemble de sommets (ou nœuds) et d’arêtes.

On note $G = (S, A)$ ou $G = (V, E)$ en anglais, le graphe défini par l’ensemble fini $S = \{S_1, S_2, \ldots, S_n\}$ dont les éléments sont appelés sommets (vertices en anglais), et par l’ensemble fini $A = \{A_1, A_2, \ldots, A_m\}$ dont les éléments sont appelés arêtes (edges en anglais).

Notions de base

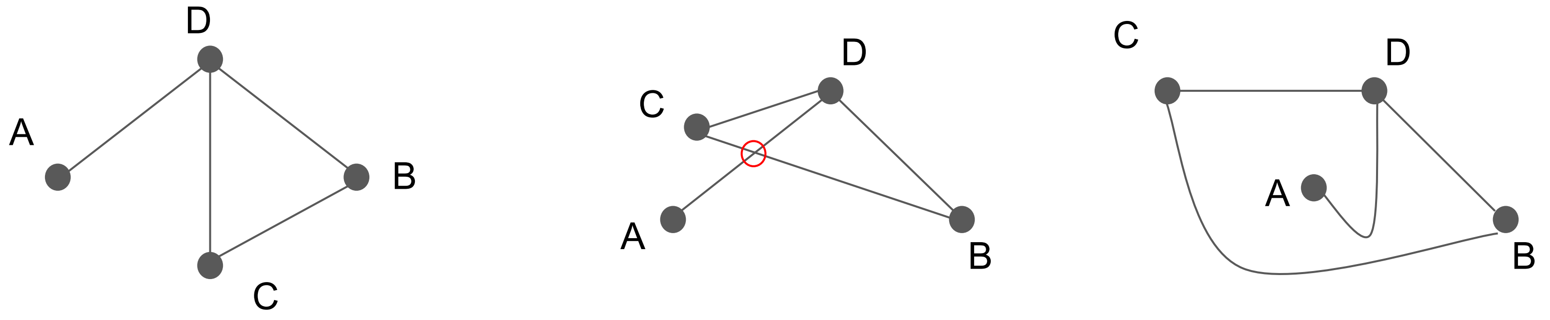
- Une **arête** ou **arc** est définie par une paire non ordonnée de sommets, appelés les *extrémités* de cette arête.
- Dans un graphe, il y a zéro ou une arête entre chaque paire de sommets.



Dessin d’un graphe

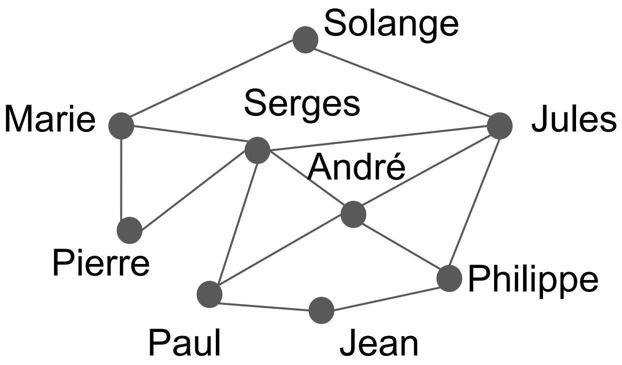
Le dessin d’un graphe est la représentation des relations entre les éléments d’un graphe. Dans les trois graphes ci-dessous on a les sommets: A, B, C, D et les arêtes: AD, BD, BC, CD.

Ces trois graphes sont les mêmes et l’intersection entre les arêtes au graphe 2 ne cause pas de problème.



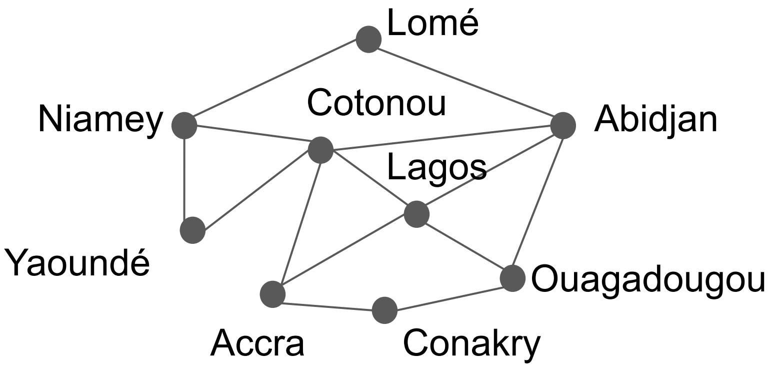
A quoi sert un graphe ?

Un graphe sert essentiellement à représenter les *relations* entre des éléments, entre ses sommets. Par exemple on peut représenter un réseau social ou les arêtes représente la relation amicale:



A quoi sert un graphe ?

On peut aussi représenter un réseau de transport ou les arêtes représentent des liaisons aériennes entre les villes (sommets):



Où trouve-t-on les graphes ?

Dans la vraie vie, les graphes sont utilisés dans les domaines suivants:

- Les réseaux sociaux, réseaux routiers, réseaux de distribution de biens;
- Relations entre des données, des objets en informatique, routage dans les réseaux;
- En bioinformatique (relations entre les séquences nucléotidiques, etc.).

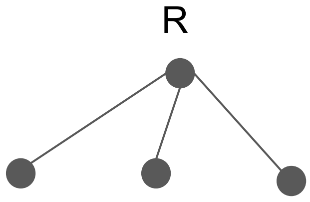
02

Notions sur les graphes

Sommets voisins et degré

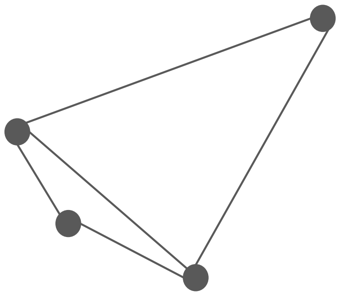
On dit que deux sommets sont **voisins** ou **adjacents** s'ils sont reliés par une arête. On peut aussi dire que l'arête est **incidente** avec les sommets.

Le **degré** d'un sommet, c'est le nombre de voisins d'un sommet. Le degré d'un graphe est le degré maximum de tous ses sommets. Par exemple, le sommet R ci-dessous a pour degré 3: deg(R) = 3.



Ordre d’un graphe

On appelle **ordre** d’un graphe le nombre de sommets de ce graphe. L’ordre du graphe ci-dessous est 4.

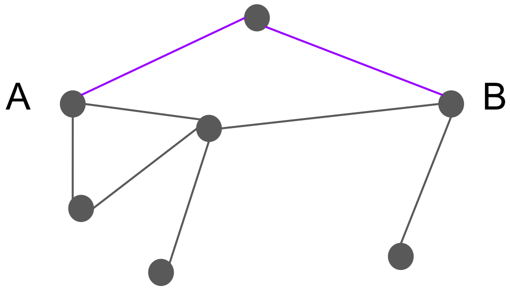


Chemin

Un chemin c'est une suite d'arête qui permet de relier deux sommets.

La longueur du chemin est égale au nombre d'arêtes du chemin.

Le chemin entre les sommets A et B sur la figure ci-contre est de longueur 2.



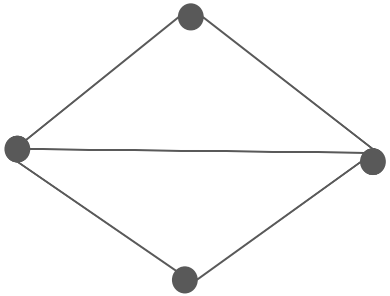
Graphe simple, non orienté et orienté

Un graphe est dit simple lorsqu’aucun sommet n’est adjacent à lui-même. Dans notre cours, nous nous restreindrons à l’étude des graphes simples.

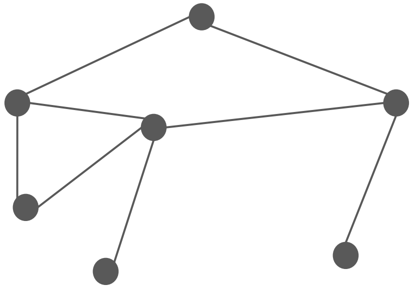
Un graphe orienté est un graphe dont les arêtes sont orientées, à savoir qu'il est possible de distinguer l'extrémité initiale d'une arête de son extrémité finale. Dans ce cas, on parlera d'arc plutôt que d'arête.

Graphe complet

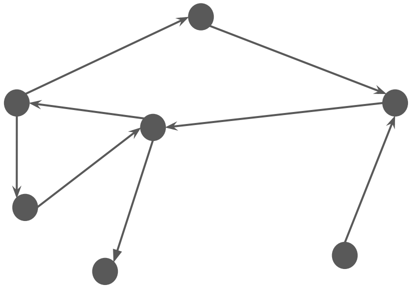
C’est un graphe dans lequel toutes les arêtes possibles entre les sommets est représenté.



Graphe simple, non orienté et orienté



Graphe non orienté

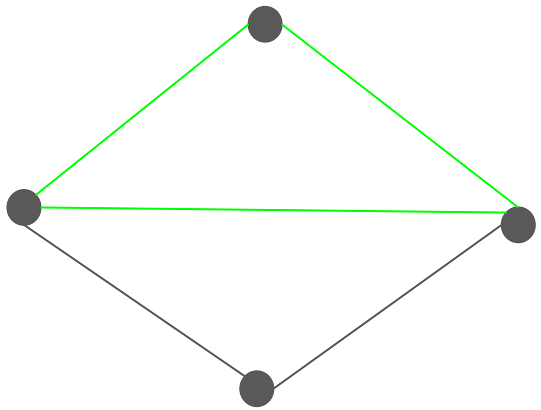


Graphe orienté

Cycle ou boucle

Un cycle c'est un chemin dont les deux extrémités sont reliées. Sa longueur est égale au nombre d'arêtes.

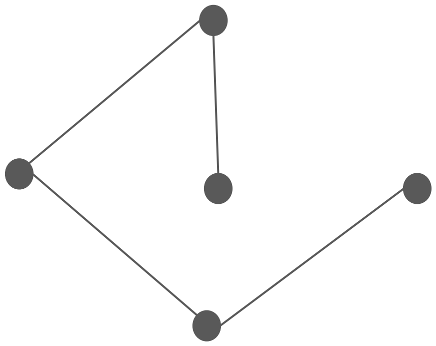
Le cycle en vert ci-dessous a pour longueur 3.



Arbre

Un arbre c’est un graphe connexe qui ne contient pas de cycle.

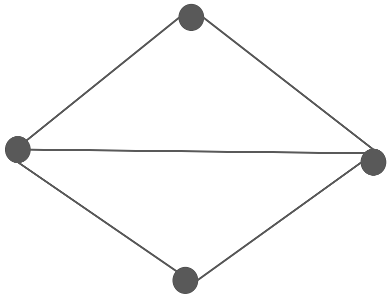
Un arbre couvrant est un arbre qu’on obtient à partir d’un graphe après avoir retiré certaines arêtes.



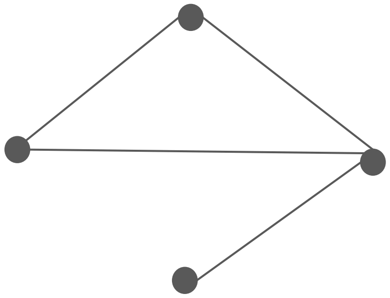
Arbre

Connexité d’un graphe

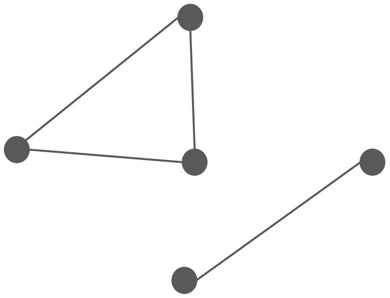
Un graphe est dit connexe si pour tout u et v (deux sommets du graphes), le graphe contient un chemin entre u et v.



Graphe connexe



Graphe connexe



Graphe non connexe

Connexité d’un graphe

Un graphe est connexe si et seulement si il contient un arbre couvrant.

Somme des degrés d’un graphe

La somme des degrés d’un graphe est égale à deux fois le nombre d’arêtes de ce graphe.

03

Modélisation de graphe

Matrice d’adjacence

Une des possibilités pour représenter un graphe est d'utiliser ce qu'on appelle une matrice d'adjacence. Dans ce type de représentation, les sommets sont **ordonnés**, et considérés comme étiquetés par des entiers de 0 à n - 1, où n est l'ordre du graphe.

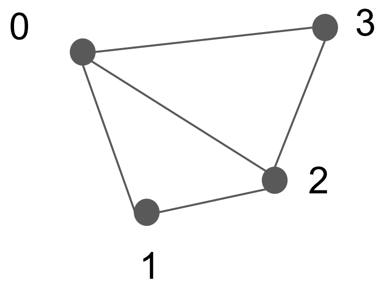
Dans cette représentation, le coefficient a_{ij} de la matrice vaut :

- 1 si il n'existe pas d'arc entre les sommets i et j;
- 0 ou la pondération s'il existe un arc entre les sommets i et j.

Matrice d’adjacence

La matrice M correspond au graphe suivant, avec les sommets dans l'ordre numérique.

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$



Représentation des matrices d’adjacence

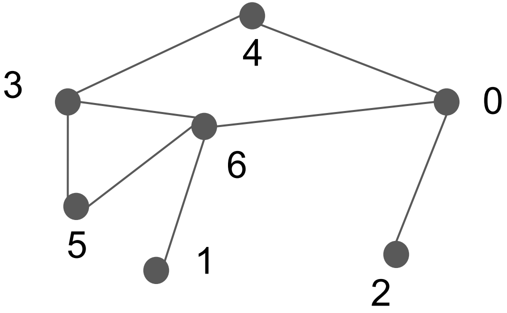
Les matrices d’adjacence sont représentées en Python par un tableau à deux dimensions (liste de liste).

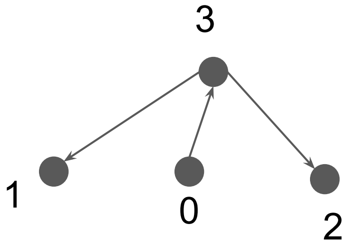
Avec un tel choix :

- Il est facile de supprimer ou d’ajouter un arc entre deux sommets existants.
- On teste en O(1) si deux sommets sont voisins.
- Ajouter un sommet nécessite en général une copie de la matrice (complexité quadratique).

Exercices

Ecrivez les matrices d'adjacence des graphes suivants:





Exercices

Tracer un graphe pouvant correspondre aux matrices suivantes. Est-ce un graphe orienté ou non-orienté ?

$$M_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \qquad M_2 = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Quelle propriété semble posséder les matrices d'adjacence d'un graphe non-orienté ?

Liste d'adjacence

Soit G = (V, A) un graphe fini simple.

On appelle liste d'adjacence de G toute liste de couples (s, l) où s parcourt V et l est une liste de ses voisins.

Si une numérotation des sommets est choisie, on peut se contenter de donner la liste des voisins. La première liste donne les voisins du premier sommet, la seconde celle du second sommet etc...

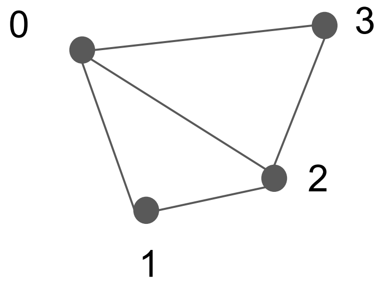
Liste d’adjacence

La liste L du graphe suivant, avec les sommets dans l'ordre numérique.

```
M = [(0, [1, 2, 3]), (1, [0, 2]), ([2, [0, 1, 3]), (3, [0, 2])]
```

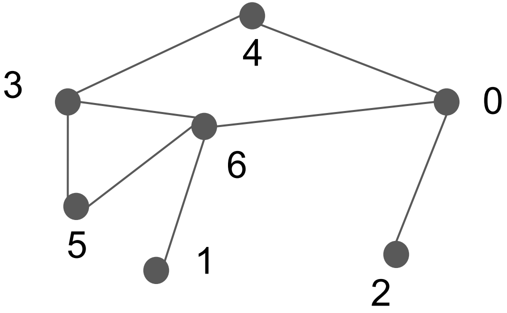
ou

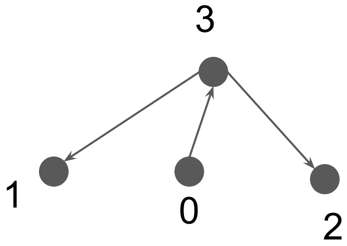
```
M = [[1, 2, 3], [0, 2], [0, 1, 3], [0, 2]]
```



Exercices

Ecrivez les listes d'adjacence des graphes suivants:





Implémentation par une liste

On peut considérer une liste L de longueur |V| de tuples (s, l) ou s est un sommet et l la liste des voisins de s.

Avantages:

- Pas de place mémoire perdue;
- Possibilité d'ajouter un nouveau sommet après avoir vérifié que ce sommet n'est pas déjà dans la liste.

Inconvénients:

- Accès à la liste d'adjacence de s en O(|V|);
- Test de voisinage entre s et x en O(|V| + deg s);
- Ajout d'un arc (s, x) en O(|V| + deg s).

Implémentation par une liste

On peut préférer gérer un tableau de listes l plutôt qu'une liste de tuples (les sommets sont alors des nombres).

Avantages:

- l'accès à la liste d'adjacence de s est en O(1) ;
- test de voisinage avec x en O(deg s) ;
- ajout d'un arc (s, x) en O(deg s) (il faut vérifier que l'arc n'est pas déjà présent).

Inconvénient: pour ajouter un sommet, il faut recopier le tableau.

Implémentation par une liste

```
graphe = [[1, 2, 3], [0, 2], [0, 1, 3], [0, 2]]

# graphe = [(0, [1, 2, 3]), (1, [0, 2]), ([2, [0, 1, 3]), (3, [0, 2])]
```

Implémentation par un dictionnaire

En partant de la liste d’adjacence sous forme de tuple on peut créer un dictionnaire dont les clés seront les sommets du graphes et les valeurs les listes des voisins de chaque sommet.

Avantages :

- Réduction de la taille en mémoire.
- Rapidité d’accès aux valeurs O(1).

Implémentation par un dictionnaire

```
graphe = {
    0: [1, 2, 3],
    1: [0, 2],
    2: [0, 1, 3],
    3: [0, 2]
}
```

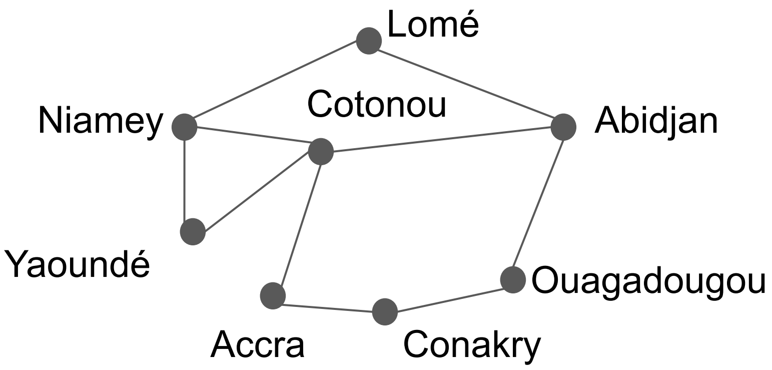
04

Pondération d’un graphe

Pondération d'un graphe

Dans le graphe ci-contre, les arêtes sont considérées comme ayant la même longueur.

Cependant pour le graphe ci-contre, il est clair que la distance entre Abidjan et Lomé est différente de la distance entre Lomé et Niamey par exemple.



Pondération d’un graphe

On peut donc préciser les distances entre les villes sur le graphe.

On parlera du **poids d’une arête.** On dit alors que le graphe est pondéré.