

Graphes : Vocabulaire, Représentation

Yves Josse

Lycée Chateaubriand – PCSI 3

Année 2023-2024

Définition d'un graphe

Vocabulaire

Un graphe G est un schéma contenant des points appelés sommets (ou nœuds) reliés ou non par des arêtes (ou segments ou liens)

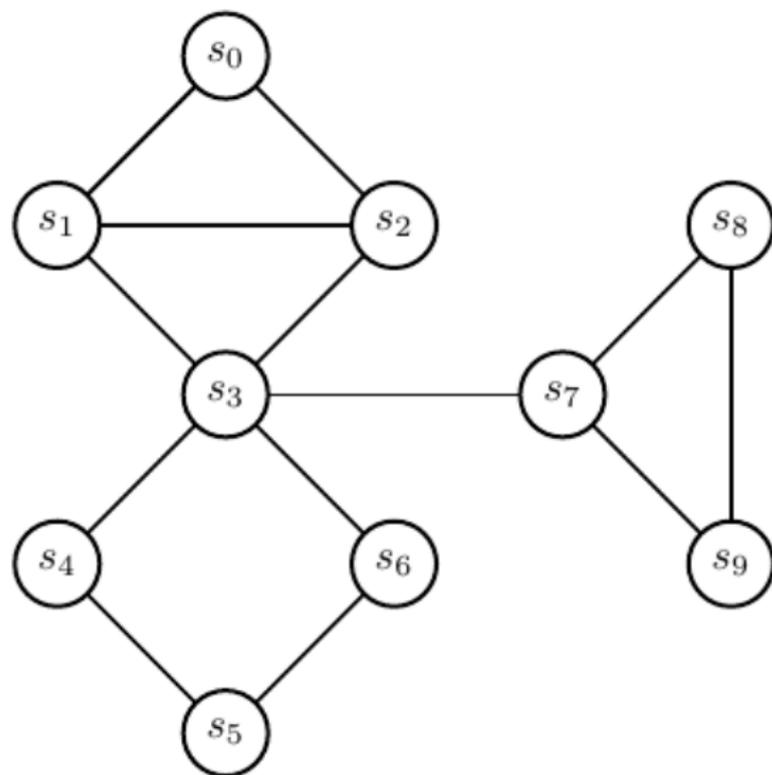
- On utilise la notation $G = (S, A)$ où G est un couple d'ensemble finis, dont :
 - S est l'ensemble des sommets de G ;
 - A est l'ensemble des arêtes de G .
- Si une arête relie les sommets s et s' , les sommets s et s' sont voisins ou adjacents.
- L'ordre d'un graphe est le nombre total de sommets.
- Une boucle est une arête reliant un sommet à lui-même.
- Un graphe est orienté si les arêtes sont orientés : on ne peut les parcourir que dans un sens.

Exemples : Notion de réseaux (routier, eau, électricité, social internet), jeux, labyrinthe, arbres. . .

Vocabulaire : graphes non orientés

- **taille** d'un graphe : nombre total d'arêtes.
- **degré** d'un sommet, noté $d(s)$: nombre d'arêtes dont ce sommet est une extrémité. Dans le cas d'une boucle elle est comptée deux fois.
- **chaîne** : suite d'arêtes consécutives reliant un sommet s à un sommet s' .
- **longueur d'une chaîne** : nombre d'arêtes de la chaîne. Pour un graphe pondéré cela correspond à la somme des poids des arêtes qui le constituent.
- **distance** entre deux sommets s et s' : longueur de la plus courte chaîne reliant s à s' . Si absence de chaîne entre s et s' : $dist(s, s') = \infty$.
- **chaîne simple** : toutes les arêtes de la chaîne sont différentes.
- **cycle** : chaîne simple où le sommet d'arrivée est le sommet de départ.
- s' est **accessible** à partir d'un sommet s s'il existe une chaîne reliant s et s' .
- **graphe connexe** : graphe où tous les sommets sont accessibles (pas de sommet isolé).

Exemple d'un graphe non orienté



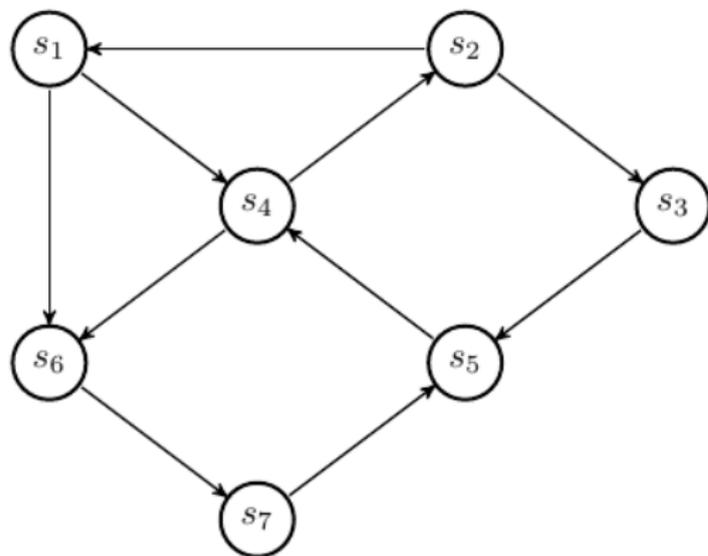
Questions :

- taille du graphe ?
- $d(s_3)$?
- $dist(s_0, s_5)$?
- nombre de chaînes simples entre s_0 et s_9 ?
- s_8 accessible depuis s_0 ?
- nombre de cycles ?
- graphe connexe ?

Vocabulaire : graphes orientés

- **arc** : arête orienté : $s \rightarrow s'$. s' est le successeur de s , s le prédécesseur de s' .
- **taille** d'un graphe orienté : nombre total d'arcs.
- **degré sortant** de s : $d_+(s)$, nombre d'arcs dont s est le point de départ.
- **degré entrant** de s : $d_-(s)$, nombre d'arcs dont s est le point d'arrivée.
- **degré** : $d(s) = d_+(s) + d_-(s)$.
- **chemin** entre s et s' : suite d'arcs consécutifs entre s et s' .
- **longueur d'un chemin** : nombre d'arcs du chemin (somme des poids si arcs pondérés).
- **distance** entre deux sommets s et s' : longueur du plus court chemin reliant s à s' . Si absence de chemin entre s et s' : $dist(s, s') = \infty$.
- **chemin simple** : tous les arcs du chemin sont différents.
- **circuit** : chemin simple où le sommet d'arrivée est le sommet de départ.

Exemple d'un graphe orienté



Questions :

- taille du graphe ?
- $d_+(s_1)$, $d_-(s_6)$, $d(s_4)$?
- $dist(s_1, s_5)$?
- nombre de chemins simples entre s_1 et s_4 ?
- s_1 accessible depuis s_6 ?
- nombre de circuit ?

Représentation d'un graphe : Liste d'adjacence

Cas d'un graphe non orienté

On représente un graphe non orienté en précisant pour chacun de ses sommets la liste de ses voisins. Ces listes s'appellent des listes d'adjacence. L'ordre d'écriture n'a pas d'importance. En python on peut utiliser des listes ou des dictionnaires (les clés sont les sommets et les valeurs les clés des voisins).

Exemples : $GL = [[\text{'s1'}, [\text{'s2'}, \text{'s5'}]], [\text{'s2'}, [\text{'s1'}, \text{'s3'}, \text{'s5'}]], [\text{'s3'}, [\text{'s2'}]], [\text{'s4'}, []], [\text{'s5'}, [\text{'s1'}, \text{'s2'}]]]$

$GD = \{\text{'s1'}: [\text{'s2'}, \text{'s5'}], \text{'s2'}: [\text{'s1'}, \text{'s3'}, \text{'s5'}], \text{'s3'}: [\text{'s2'}], \text{'s4'}: [], \text{'s5'}: [\text{'s1'}, \text{'s2'}]\}$

Cas d'un graphe orienté

Dans le cas d'un graphe orienté, une représentation par liste d'adjacence consiste à associé à chaque sommet la liste de ses successeurs. Sous python, on peut utiliser une liste de listes ou un dictionnaire.

Exemple d'un graphe orienté pondéré :

$GOP = \{\text{'s1'}: [(\text{'s2'}, 2), (\text{'s5'}, 5)], \text{'s2'}: [(\text{'s1'}, 2), (\text{'s3'}, 1), (\text{'s5'}, 3)], \text{'s3'}: [(\text{'s2'}, 1)], \text{'s4'}: [], \text{'s5'}: [(\text{'s1'}, 5), (\text{'s2'}, 3)]\}$

Représentation d'un graphe : Matrice d'adjacence

Matrice d'adjacence

On peut aussi représenter un graphe par sa matrice d'adjacence : on numérote les sommets de 1 à n et on obtient une matrice carrée de taille n en mettant un 1 en position (i, j) s'il y a un arc ou une arête joignant le sommet i au sommet j et 0 sinon.

- Pour un graphe non orienté, la matrice d'adjacence est symétrique avec des 0 sur la diagonale (si on exclut les boucles).
- Dans le cas d'un graphe pondéré (ou étiqueté), on remplace les 1 de la matrice par les poids.

Représentation en python

La matrice d'adjacence est implémenter sous python par une liste de listes

Exemple 1 : Graphe orienté :

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

Exemple 2 : Graphe orienté et pondéré :

$$M_2 = \begin{pmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 4 \\ 0 & 3 & 0 & 0 \\ 3 & 1 & 0 & 0 \end{pmatrix}$$

Voisins, degré, longueur

On considère un graphe pondéré G non orienté sans boucle défini par sa matrice d'adjacence $M_{i,j}$.

- 1 Écrire une fonction `voisins(M,i)` renvoyant la liste des voisins du sommet i .
- 2 Écrire une fonction `degre(M,i)` renvoyant le nombre de voisins du sommet i .
- 3 Écrire une fonction `longueur(M,L)` où L est une liste de sommets de G renvoyant la longueur de la chaîne décrit par la liste. Si le trajet n'est pas possible, la fonction renverra -1 .

Conversions

On dispose d'un graphe non orienté implémenté sous la forme de listes d'adjacence par un dictionnaire g .

- 1 Écrire une fonction `conversion1(g)` qui prend en paramètre le dictionnaire et qui renvoie la matrice d'adjacence M correspondante.
- 2 Écrire une fonction `conversion2(M)` qui prend en paramètre la matrice d'adjacence et qui renvoie un dictionnaire où le graphe est implémenté par une liste d'adjacence.