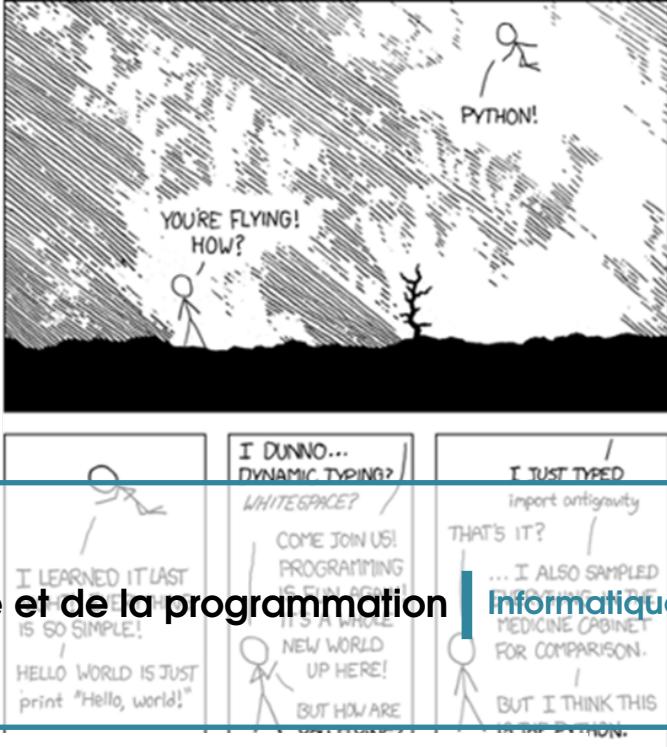
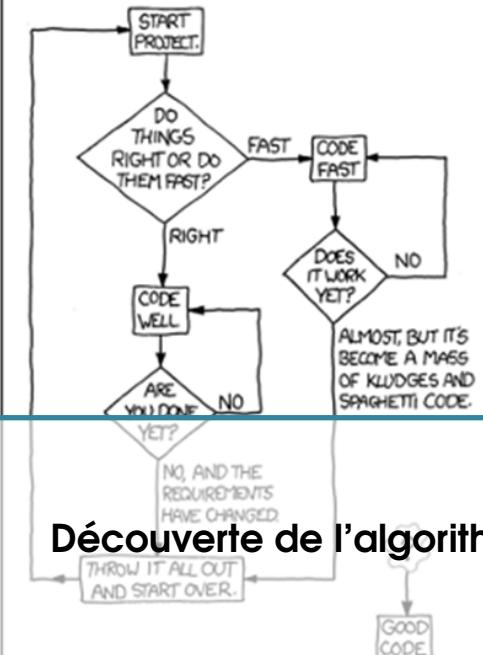


HOW TO WRITE GOOD CODE:



Découverte de l'algorithmatique et de la programmation

Chapitre 11

Introduction aux graphes

Savoirs et compétences :

Cours

- Vocabulaire des graphes : graphe orienté, graphe non orienté. Sommet (ou nœud); arc, arête. Boucle. Degré (entrant et sortant). Chemin d'un sommet à un autre. Cycle. Connexité dans les graphes non orientés.
- Notations : graphe $G = (S, A)$, degrés $d(s)$ (pour un graphe non orienté), $d_+(s)$ et $d_-(s)$ (pour un graphe orienté).
- Pondération d'un graphe. Étiquettes des arcs ou des arêtes d'un graphe. On motive l'ajout d'information à un graphe par des exemples concrets.

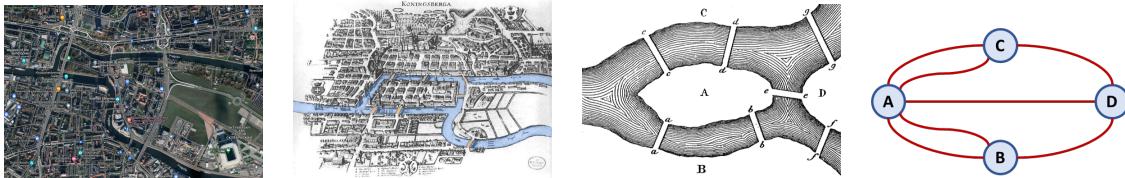


Représentation circulaire du métro parisien

1	Introduction	2
2	Vocabulaire des graphes	2
3	Chemins	3
4	Implémentation des graphes	5

1 Introduction

Historiquement, le problème des sept ponts de Königsberg (Kaliningrad) était de déterminer s'il existe une promenade qui partait de n'importe quel quartier et qui y revenait, tout en passant une et une seule fois par chacun des ponts. Euler mis en forme le problème et le résolut... en montrant que dans ce cas, il n'existe pas de solution.



Les applications des graphes sont assez nombreuses :

- recherche d'itinéraire pour aller d'un point à un autre;
- routage des paquets sur Internet;
- analyse des réseaux sociaux...

2 Vocabulaire des graphes

Définition Graphe Un graphe est un ensemble de **sommets** et **relations** entre ces sommets.

Lorsque deux sommets sont en relation, on dit qu'il existe une **arête** entre ces sommets.

Définition Graphe non orienté – Arêtes Un graphe non orienté G est un couple $G = (S, A)$, où S est un ensemble fini de sommets (appelés aussi noeuds) et où A est un ensemble fini de paires ordonnées de sommets, appelées arêtes.

On note $x - y$ l'arête $\{x, y\}$. x et y sont les deux extrémités de l'arête.

Définition Graphe orienté – Arcs [ref_01] Un graphe orienté G est un couple $G = (S, A)$, où S est un ensemble fini de sommets et où A est un ensemble fini de paires ordonnées de sommets, appelées arcs.

On note $x \rightarrow y$ l'arc (x, y) . x est l'extrémité initiale de l'arc, y est son extrémité terminale. On dit que y est successeur de x et que x est prédécesseur de y .

■ Exemple

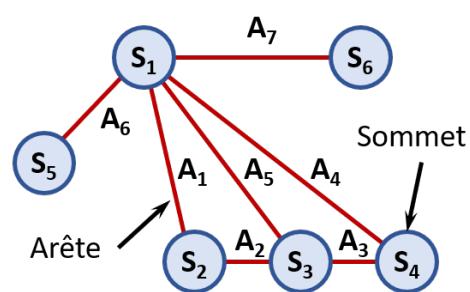


FIGURE 1 – Graphe non orienté

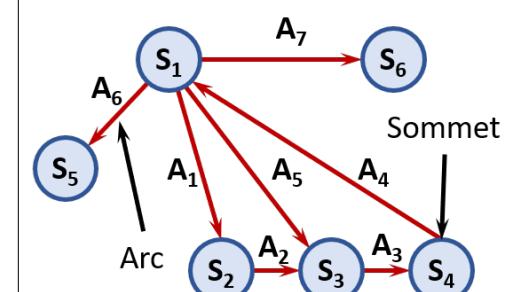


FIGURE 2 – Graphe orienté



On peut noter le graphe non orienté $G = ([1, 6], E)$ où $E = (\{1, 2\}, \{2, 3\}, \{3, 4\}, \{1, 4\}, \{1, 3\}, \{1, 5\}, \{1, 6\})$ désigne les arêtes.

On peut noter le graphe orienté $G = ([1, 6], E)$ où $E = ((1, 2), (2, 3), (3, 4), (1, 4), (1, 3), (1, 5), (1, 6))$ désigne les arcs.

Définition Adjacence Deux arcs (resp. arêtes) d'un graphe orienté (resp. non orienté) sont dits adjacents s'ils ont au moins une extrémité commune.

Deux sommets d'un graphe non orienté sont dits adjacents s'il existe une arête les joignant.

Dans un graphe orienté, le sommet y est dit adjacent au sommet x s'il existe un arc $x \rightarrow y$.

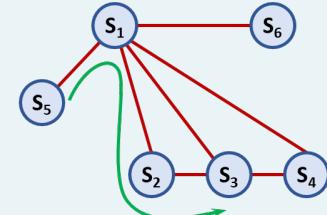
Définition Graphes pondérés Étiqueter les arêtes d'un graphe (S, A) (orienté ou non), c'est se donner une fonction $f : A \rightarrow V$ (où V est un ensemble de valeurs). On dit qu'un graphe est pondéré si ses arêtes sont étiquetées par des nombres. On parlera alors du poids d'une arête.

3 Chemins

Définition Chemin dans un graphe

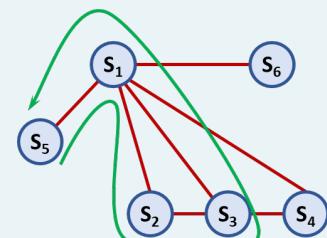
On appelle chemin dans un graphe une suite finie $\{S_0, \dots, S_{n-1}\}$ de n sommets tels que pour tout $i \in \llbracket 0, n-1 \rrbracket$, une arête relie S_i à S_{i+1} . On dit que ce chemin relie le sommet de départ S_0 au sommet de fin S_{n-1} .

Dans le cas d'un graphe non orienté, les arêtes sont notées $\{S_i, S_{i+1}\}$ pour $i \in \llbracket 0, n-1 \rrbracket$. Dans le cas d'un graphe orienté, les arcs sont notées (S_i, S_{i+1}) pour $i \in \llbracket 0, n-1 \rrbracket$.



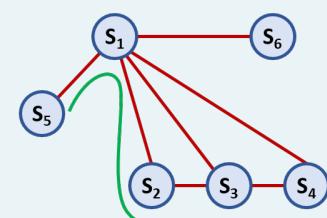
Définition Chemin fermé

Chemin dont le sommet de départ et le sommet d'arrivée sont identiques.



Définition Chemin élémentaire

Chemin n'empruntant que des arêtes distinctes.



Définition Chemin simple Chemin tel que les $n-2$ sommets intermédiaires si, pour $i \in \llbracket 1, n-1 \rrbracket$ soient deux à deux distincts et tous distincts du sommet de départ S_0 et du sommet d'arrivée S_{n-1} et tels que ce chemin n'est pas de la forme a, b, a dans le cas non-orienté.

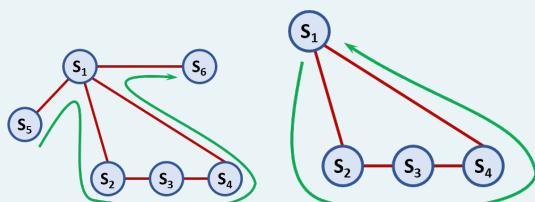
Définition Circuit Chemin fermé de longueur non nulle.

Définition Cycle Circuit élémentaire (chemin fermé de longueur non nulle dont toutes les arêtes sont distinctes).

Définition Cycle simple Chemin fermé et simple de longueur non nulle.

Définition Chemin et cycle eulérien

Chemin (resp. cycle) contenant une et une seule fois toutes les arêtes du graphe.



Pour certains auteurs, un chemin élémentaire est ce que nous avons appelé un chemin simple et réciproquement. Pour d'autres, un cycle est ce que nous avons appelé un cycle simple.

Définition Connexité dans les graphes non orientés Un graphe $G = (S, A)$ est dit connexe si, pour deux sommets quelconques S_i et S_j de S , il existe un chemin de S_i à S_j .

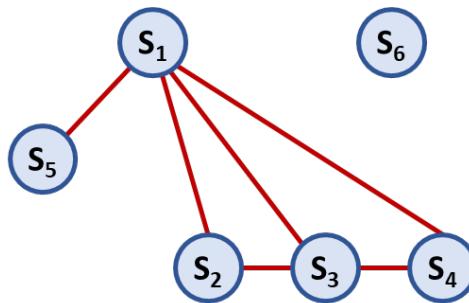


FIGURE 3 – Graphe ayant 2 composantes connexes

■ Exemple

3.1 Notations

Définition Degré d'un sommet On appelle degré d'un sommet s et on note $d(s)$ le nombre d'arcs (ou d'arêtes) dont s est une extrémité.

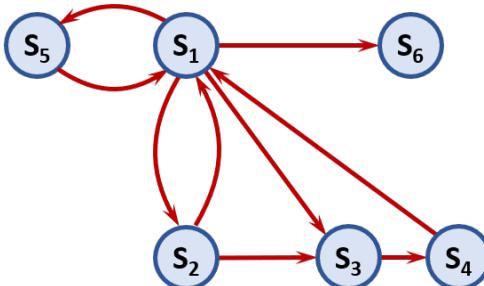
Définition Degré entrant et sortant On note s le sommet d'un graphe orienté. On note :

$d_+(s)$ le demi-degré extérieur de s , c'est-à-dire le nombre d'arcs ayant leur extrémité initiale en s (ces arcs sont dits incidents à s vers l'extérieur);

- $d_-(s)$ le demi-degré intérieur de s , c'est-à-dire le nombre d'arcs ayant leur extrémité finale en s (ces arcs sont dits incidents à s vers l'intérieur).

Dans ce cas, on a $d^\circ(s) = d_-(s) + d_+(s)$.

■ Exemple



- $d_-(S_1) = 3$.
- $d_+(S_1) = 4$.
- $d^\circ(S_1) = 7$.

FIGURE 4 – Graphe orienté

4 Implémentation des graphes

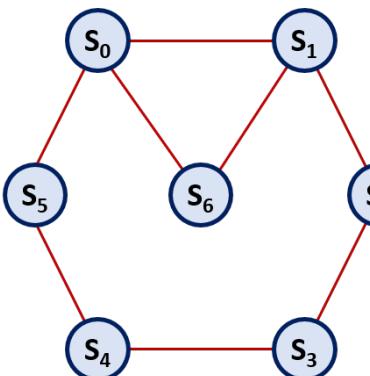
4.1 Liste d'adjacence

Définition Liste d'adjacence Soit un graphe de n sommets d'indices $i \in [0, n - 1]$. Pour implémenter le graphe, on utilise une liste G de taille n pour laquelle, $G[i]$ est la liste des voisins de i .



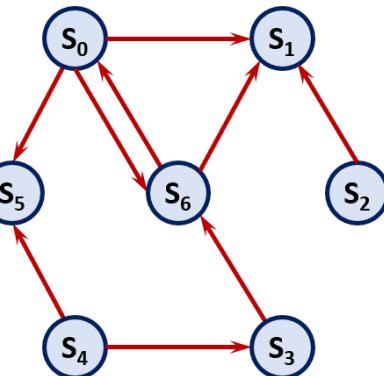
Cette implémentation est plutôt réservée au graphes « creux », c'est-à-dire ayant peu d'arêtes.

■ Exemple



Dans ce cas S_0 est voisin de S_1 , S_5 et S_6 ; donc $G[0] = [1, 5, 6]$. S_3 est voisin de S_2 et S_4 ; donc $G[3] = [2, 4]$.

```
G = [[1,5,6], [0,2,6], [1,3], [2,4], [3,5],
     [4,0], [1,0]]
```



Dans ce cas, le graphe est orienté. La liste d'adjacence contient la liste des successeurs. Ainsi, les successeurs de S_0 sont S_1 , S_5 et S_6 ; donc $G[0] = [1, 5, 6]$. S_1 n'a pas de successeur donc $G[1] = []$.

```
G = [[1,5,6], [], [1], [6], [3,5], [], [0,1]]
```

Dans la même idée, il est aussi possible d'utiliser des dictionnaires d'adjacence dans lequel les clés sont les sommets, et les valeurs sont des listes de voisins ou de successeurs.

```
# Graphe non orienté
G = {"S0": ["S1", "S5", "S6"], "S1": ["S0", "S2", "S6"], "S2": ["S1", "S3"], "S3": ["S2", "S4"], "S4": ["S3", "S5"], "S5": ["S4", "S0"], "S6": ["S1", "S0"]}
# Graphe orienté
G = {"S0": ["S1", "S5", "S6"], "S1": [], "S2": ["S1"], "S3": ["S6"], "S4": ["S3", "S5"], "S5": [], "S6": ["S1", "S0"]}
```

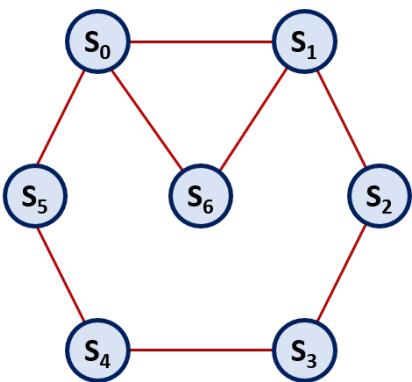
4.2 Matrice d'adjacence

Définition Matrice d'adjacence Soit un graphe de n sommets d'indices $i \in \llbracket 0, n-1 \rrbracket$ et E l'ensemble des arêtes (on notera $G = (\llbracket 0, n-1 \rrbracket, E)$). Pour implémenter le graphe, on utilise la matrice d'adjacence carrée de taille n , $\mathcal{M}_n G$ de taille n pour laquelle, $m_{i,j} = \begin{cases} \text{True} & \text{si } \{i, j\} \in E \\ \text{False} & \text{sinon} \end{cases}$ avec $i, j \in \llbracket 0, n-1 \rrbracket$.



Cette implémentation est plutôt réservée au graphes « denses » ayant « beaucoup » d'arêtes.

■ Exemple

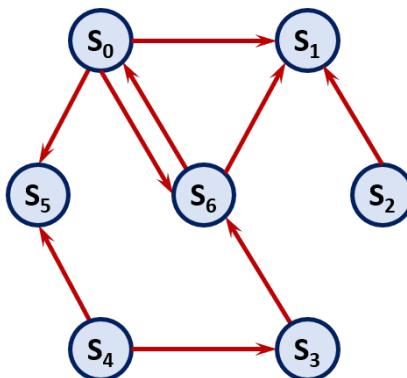


On a dans ce cas

$$M = \begin{pmatrix} \text{False} & \text{True} & \text{False} & \text{False} & \text{False} & \text{True} & \text{True} \\ \text{True} & \text{False} & \text{True} & \text{False} & \text{False} & \text{False} & \text{True} \\ \text{False} & \text{True} & \text{False} & \text{True} & \text{False} & \text{False} & \text{False} \\ \text{False} & \text{False} & \text{True} & \text{False} & \text{True} & \text{False} & \text{False} \\ \text{False} & \text{False} & \text{False} & \text{True} & \text{False} & \text{True} & \text{False} \\ \text{True} & \text{False} & \text{False} & \text{False} & \text{True} & \text{False} & \text{False} \\ \text{True} & \text{True} & \text{False} & \text{False} & \text{False} & \text{False} & \text{False} \end{pmatrix}$$

ou

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



Dans ce cas, le graphe est orienté. On a dans ce cas

$$M = \begin{pmatrix} \text{False} & \text{True} & \text{False} & \text{False} & \text{False} & \text{True} & \text{True} \\ \text{False} & \text{False} & \text{False} & \text{False} & \text{False} & \text{False} & \text{False} \\ \text{False} & \text{True} & \text{False} & \text{False} & \text{False} & \text{False} & \text{False} \\ \text{False} & \text{False} & \text{False} & \text{False} & \text{False} & \text{False} & \text{True} \\ \text{False} & \text{False} & \text{False} & \text{True} & \text{False} & \text{True} & \text{False} \\ \text{False} & \text{False} & \text{False} & \text{False} & \text{False} & \text{False} & \text{False} \\ \text{True} & \text{True} & \text{False} & \text{False} & \text{False} & \text{False} & \text{False} \end{pmatrix}$$

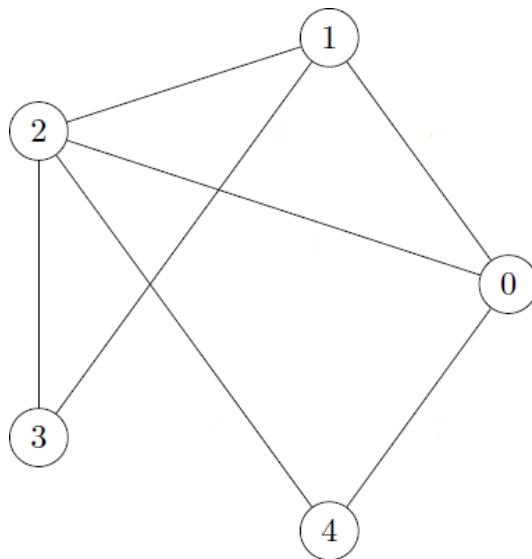
$$\text{ou } M = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

R

- Dans le cas d'un graphe non orienté, la matrice est symétrique.
- Si on avait un bouclage sur un sommet, il y aurait des valeurs non nulles sur la diagonale.

Exercice 1 – Implémentation des graphes par une liste d'adjacence

On considère le graphe G suivant, où le nombre situé sur l'arête joignant deux sommets est leur distance, supposée entière.



Pour implémenter le graphe, on utilise une liste G1 qui a pour taille le nombre de sommets. Chaque élément G1[i] est la liste des voisins de i.

Dans ce cas, G1[0]=[1, 2, 4] car Les sommets 1, 2 et 4 sont des voisins de 0.

Question 1 Construire la liste d'adjacence G1 en utilisant la méthode énoncée ci-dessus.

Question 2 Écrire une fonction voisins_1(G:list, i:int) -> list, d'argument la liste d'adjacence G et un sommet i, renvoyant la liste des voisins du sommet i.

Question 3 Écrire une fonction arretes_1(G:list) -> list, renvoyant la liste des arêtes. Les arêtes seront constitués de couples de sommets (l'arête entre les sommets 0 et 1 sera donnée par (0, 1)).

Les instructions suivantes permettent de tracer un graphe.

```
import networkx as nx

def plot_graphe_1(G):
    Gx = nx.Graph()
    edges = arretes_1(G)
    Gx.add_edges_from(edges)
    nx.draw(Gx, with_labels = True)
    plt.show()
plot_graphe(M)
```

Question 4 Écrire et tester la fonction plot_graphe_1(G).

Question 5 Écrire une fonction degré_1(G:list, i:int) -> int, d'argument un sommet i, renvoyant le nombre des voisins du sommet i, c'est-à-dire le nombre

d'arêtes issues de i.

Question 6 Écrire la fonction ajout_sommet_1(G:list, L:list) -> None permettant d'ajouter un sommet au graphe. L désigne la liste des sommets auxquels le nouveau sommet est relié. ajout_sommet agit avec effet de bord sur G.

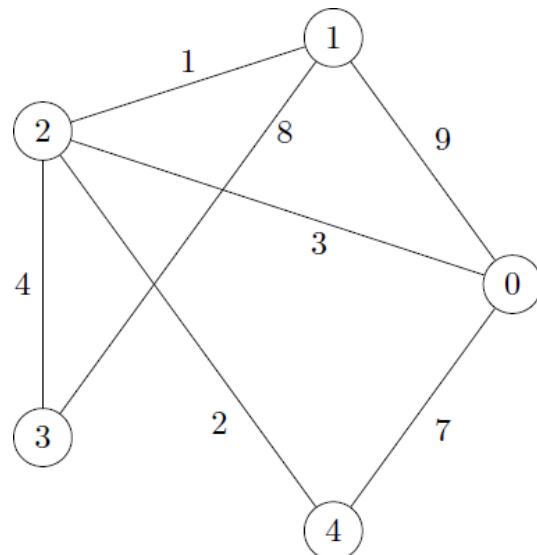
Question 7 Écrire la fonction supprime_sommet_1(G:list, i: int) -> None permettant de supprimer le sommet i du graphe.

Question 8 Écrire la fonction from_list_to_matrix(G:list) -> list permettant de convertir un graphe implémenté sous forme de liste d'adjacence en matrice d'adjacence.

Question 9 Écrire la fonction from_matrix_to_list -> list permettant de convertir un graphe implémenté sous forme de matrice d'adjacence en liste d'adjacence.

Exercice 2 – Implémentation des graphes par une matrice d'adjacence

On considère le graphe G suivant, où le nombre situé sur l'arête joignant deux sommets est leur distance, supposée entière.



Question 10 Construire la matrice $(G_{ij})_{0 \leq i,j \leq 4}$, matrice de distances du graphe G, définie par : « pour tous les indices i, j , G_{ij} représente la distance entre les sommets i et j , ou encore la longueur de l'arête reliant les sommets i et j ». Cette matrice sera implémentée sous forme d'une liste de listes. (Chaque « sous-liste » représentant une ligne de la matrice d'adjacence. On convient que, lorsque les sommets ne sont pas reliés, cette distance vaut -1 . La distance du sommet i à lui-même est égale à 0).

Question 11 Écrire une fonction voisins(G:list, i:int) -> list, d'argument la matrice d'adjacence G et un sommet i, renvoyant la liste des voisins du sommet i.

Question 12 Écrire une fonction `arêtes(G:list)` → `list`, renvoyant la liste des arêtes. Les arêtes seront constitués de couples de sommets (l'arête entre les sommets 0 et 1 sera donnée par (0, 1)).

Les instructions suivantes permettent de tracer un graphe.

```
import networkx as nx
import matplotlib.pyplot as plt

def plot_graphe(G):
    Gx = nx.Graph()
    edges = arêtes(G)
    Gx.add_edges_from(edges)
    nx.draw(Gx, with_labels = True)
    plt.show()
plot_graphe(M)
```

Question 13 Écrire et tester la fonction `plot_graphe(G)`.

Question 14 Écrire une fonction `degré(G:list,`

`i:int) -> int`, d'argument un sommet `i`, renvoyant le nombre des voisins du sommet `i`, c'est-à-dire le nombre d'arêtes issues de `i`.

Question 15 Écrire une fonction `longueur(G:list,L:list)` → `int`, d'argument une liste `L` de sommets de `G`, renvoyant la longueur du trajet écrit par cette liste `L`, c'est-à-dire la somme des longueurs des arêtes empruntées. Si le trajet n'est pas possible, la fonction renverra `-1`.

Question 16 Écrire la fonction `ajout_sommet(G:list, L:list, poids : list)` → `None` permettant d'ajouter un sommet au graphe. `L` désigne la liste des sommets (triés dans l'ordre croissant) auxquels le nouveau sommet est relié, `poids` la liste des poids respectifs. `ajout_sommet` agit avec effet de bord sur `G`.

Question 17 Écrire la fonction `supprime_sommet(G:list, i: int)` → `None` permettant de supprimer le sommet `i` du graphe.