

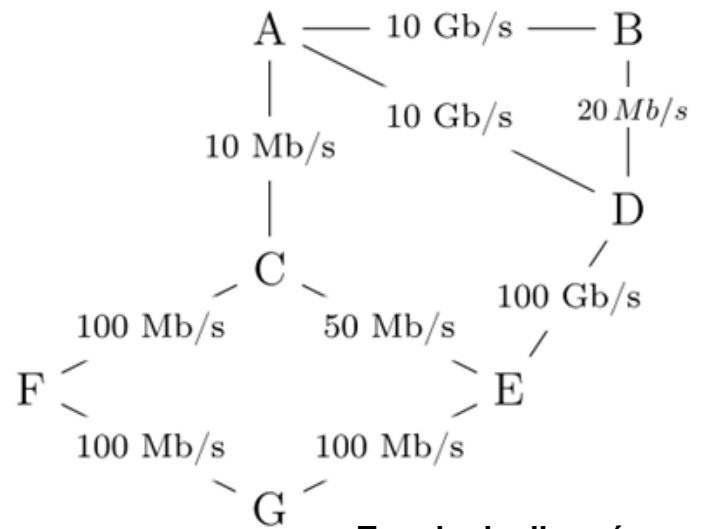
# Les graphes

## Résumé

Les graphes permettent de caractériser une représentation d'une solution à un grand nombre de problèmes de la vie courante.

Savoir utiliser un graphe pour en déduire l'existence de relations entre deux entités, l'existence d'un chemin, le coût de transit comparatif entre deux chemins est alors possible.

Il existe un très grand nombre d'applications industrielles tel que : le routage d'informations dans les réseaux informatique, l'organisation des tâches dans un atelier de production.

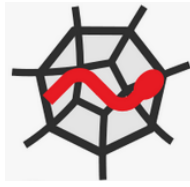


**Topologie d'un réseau**

Ce domaine est l'un des domaines actif de la recherche en algorithmique.

## Sommaire

<b>1</b>	<b>Les graphes quesaco ?</b>	<b>3</b>
1.1	Présentation	3
1.2	Notions de base autour des graphes	3
1.3	Divers types de graphes	4
1.4	Vocabulaire :	4
1.5	Quelques questions	5
	Acheminement de pétrole	5
	Quels sont les graphes identiques	6
	Tracé de graphe	6
<b>2</b>	<b>Représentation en python</b>	<b>7</b>
2.1	Représentation par matrice d'adjacence	7
	Graphes orientés	7
	Graphes pondérés	8
	Graphes non orientés	9
2.2	Les scripts Python	10
	Une classe Graphe présentation	10
	Mise en œuvre codage d'un graphe non orienté	11
	Amélioration impression de la matrice d'adjacence	11
	Ajout de la possibilité d'impression directement par print(objet)	12
	Ajout d'une arête (arc non orienté)	12
	Graphe pondéré	12



**3 Autres ressources.....15**  
3.1 *Sur la théorie des graphes.....15*

# 1 Les graphes quesaco ?

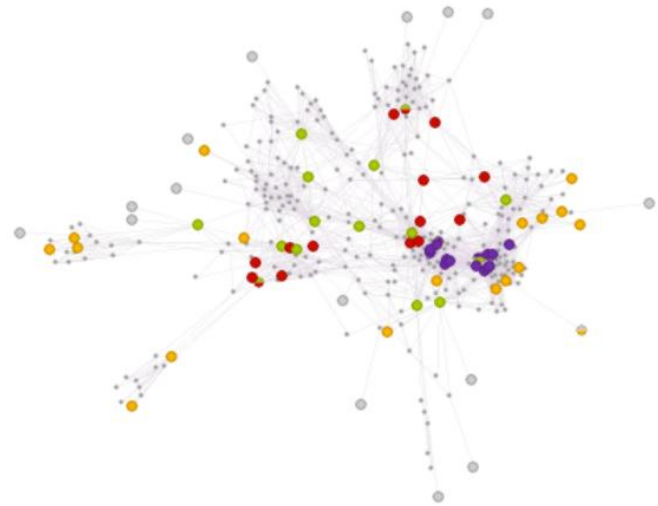
## 1.1 Présentation

Les graphes sont partout dans notre vie quotidienne, quelques exemples :



1

Réseau autoroutier



2

Le réseau d'amis de Facebook

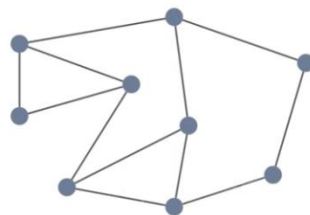
## 1.2 Notions de base autour des graphes

### Qu'est-ce qu'un graphe ?

Un ensemble de **sommets**

Un ensemble d'**arêtes**

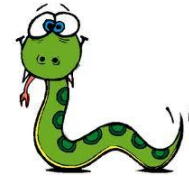
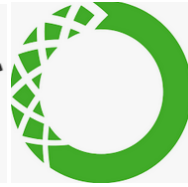
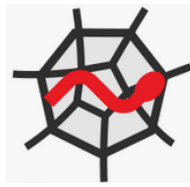
*La vidéo ci-dessous introduit les notions de bases sur les graphes.*



<https://www.youtube.com/watch?v=YYv2R1cCTa0>

<sup>1</sup> [https://static.wikia.nocookie.net/routes/images/1/10/Carte\\_autoroutes\\_Concessions.jpg/revision/latest?cb=20101115200414](https://static.wikia.nocookie.net/routes/images/1/10/Carte_autoroutes_Concessions.jpg/revision/latest?cb=20101115200414)

<sup>2</sup> <https://www.blogdumoderateur.com/analyser-son-profil-facebook-avec-wolfram-alpha/>

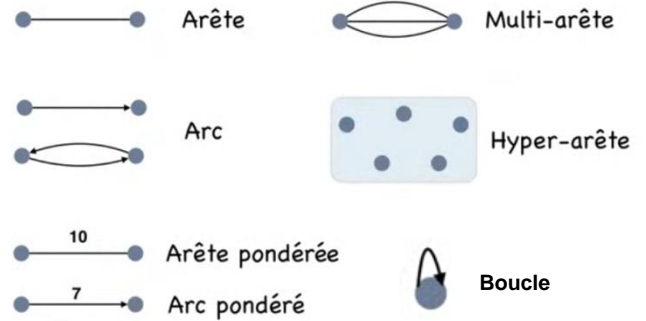


### 1.3 Divers types de graphes

Graphe orienté, non orienté, pondéré avec boucles .... tout est là :

[https://www.youtube.com/watch?v=O1YdAP-tF2U&list=RDCMUChTjVeNLyR1yuJ1\\_xCK1WRg&index=12](https://www.youtube.com/watch?v=O1YdAP-tF2U&list=RDCMUChTjVeNLyR1yuJ1_xCK1WRg&index=12)

### Diverses « arêtes »



### 1.4 Vocabulaire :

Sommets ou Nœuds

Arêtes, Arcs : graphe non orienté, graphe orienté

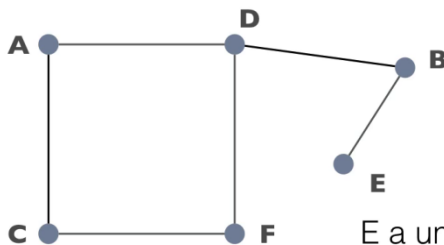
Graphe complet : il existe toutes les arêtes possibles entre tous ses sommets.

Sommets voisins : sommets reliés par une arête.

Degré d'un sommet : nombre de ses voisins.

Voisins de A : D et C

Degré de D = 3



E a un seul voisin : B

E et F ne sont **pas** voisins

Degré de E = 1

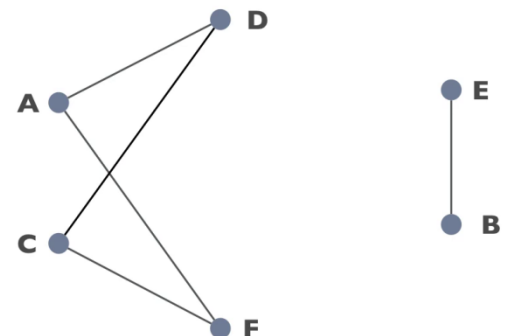
Chemin : succession d'arêtes permettant de relier deux sommets.

Longueur de chemin : le nombre d'arêtes.

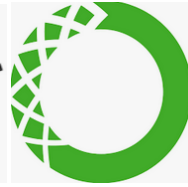
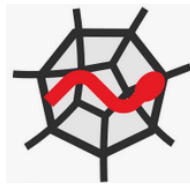
Cycle : un chemin qui part et arrive depuis le même nœud.

Graphe connexe : toute paire de sommets du graphe peut être reliée par un chemin.

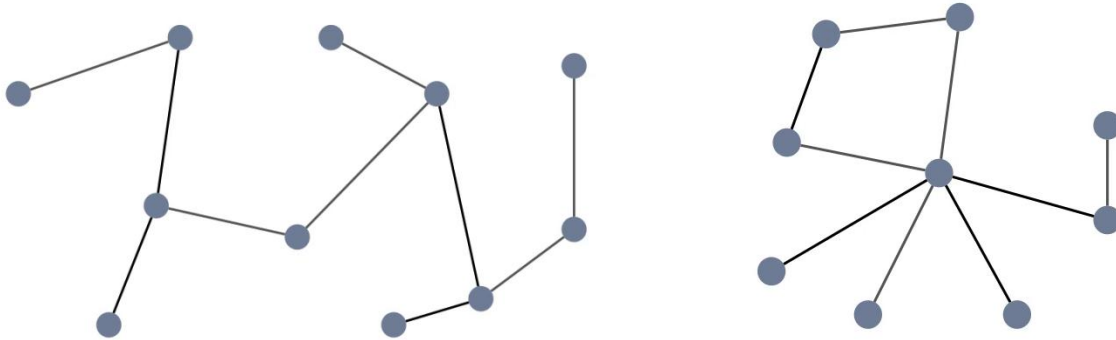
Composante connexe : graphe en plusieurs parties non reliées ensemble.



<https://www.youtube.com/watch?v=YYv2R1cCTa0>



Arbre : graphe connexe et sans cycle.



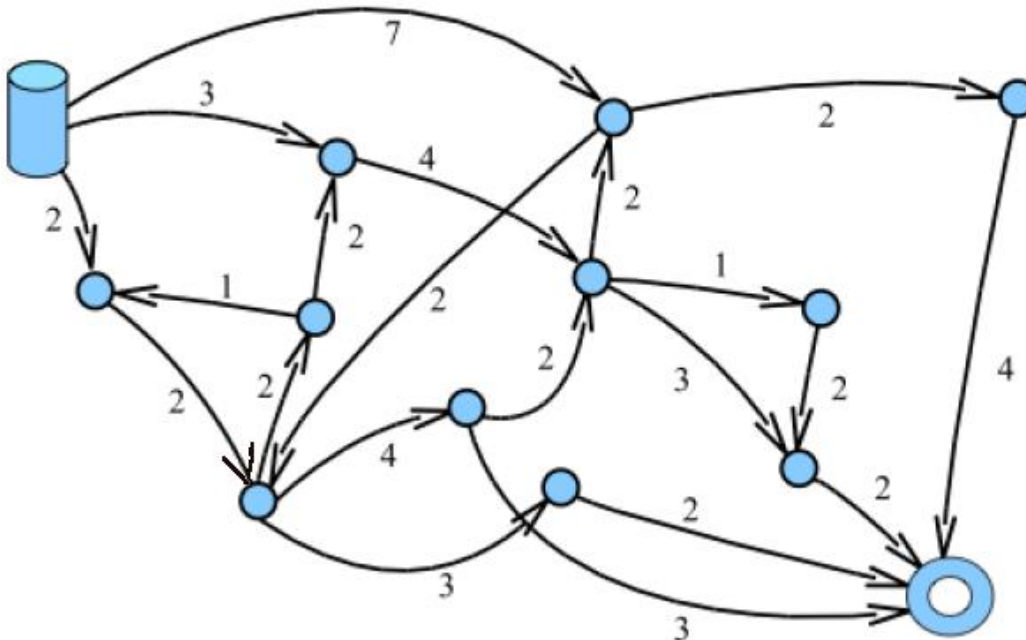
Propriété : somme des degrés =  $2 \cdot$  nombre de sommets

### 1.5 Quelques questions

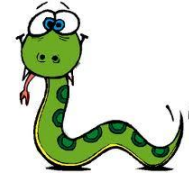
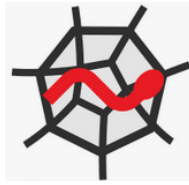
#### Acheminement de pétrole<sup>3</sup>

La compagnie pétrolière Inti T'schouff souhaite acheminer du pétrole par oléoduc vers un pays client. Le réseau d'oléoduc comporte plusieurs tronçons, chacun ayant une capacité maximale (en débit) à ne pas dépasser. Les tronçons sont directionnels. Sur le graphe suivant, la compagnie pétrolière est représentée par le cylindre, le client par le jeton. La capacité maximale de chaque arc est indiquée.

**Q1.** Quel est le débit maximum que la compagnie pétrolière peut envoyer vers le client via le réseau?

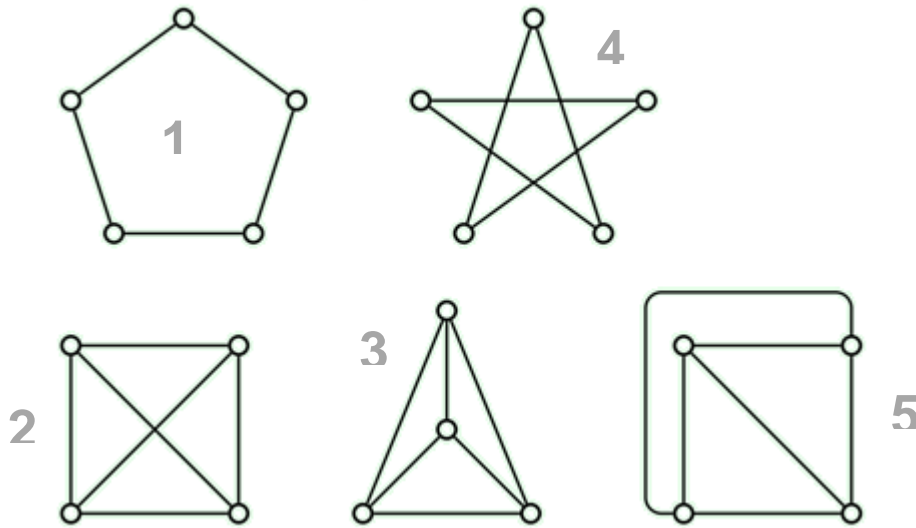


<sup>3</sup> Exercices de graphes, Nadia Brauner, UGA.



Quels sont les graphes identiques<sup>4</sup>

**Q2.** Déterminer les graphes identiques.



Tracé de graphe

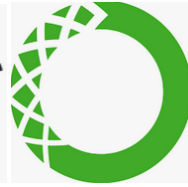
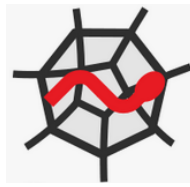
**Q3.** Construire un graphe orienté dont les sommets sont les entiers compris entre 1 et 12 et dont les arcs représentent la relation « être diviseur de »<sup>5</sup>.



<sup>4</sup> Ibid.

<sup>5</sup> D'après exo\_graphes\_sopena\_tout.pdf Éric SOPENA - [sopena@labri.fr](mailto:sopena@labri.fr)



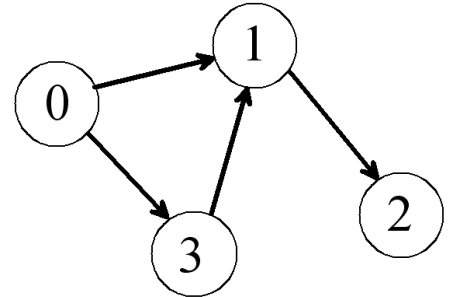


## 2 Représentation en python<sup>6</sup>

### 2.1 Représentation par matrice d'adjacence

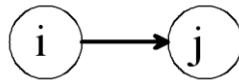
#### Graphes orientés

Voilà un graphe orienté :



Pour le représenter informatiquement on peut utiliser un tableau de booléens de dimension  $N \times N$  ou  $N$  est le nombre de nœuds ou sommets du graphe. Chaque valeur de ce tableau représente l'existence d'un arc orienté entre le nœud  $i$  et le nœud  $j$  :

- $adjacence[i][j] = True$  il existe un arc entre les nœuds  $i$  et  $j$



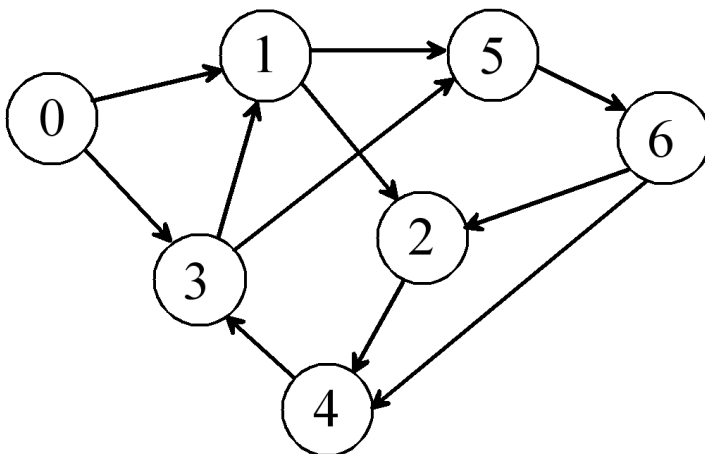
- $adjacence[i][j] = False$  il existe aucun arc entre les nœuds  $i$  et  $j$



Matrice d'adjacence du graphe ci-dessus  $\Rightarrow$

	0	1	2	3
0	F	T	F	T
1	F	F	T	F
2	F	F	F	F
3	F	T	F	F

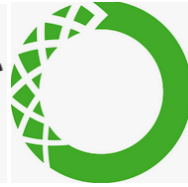
**Q4.** Donner la matrice d'adjacence du graphe ci-dessous :



	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

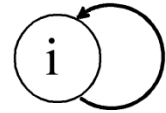


<sup>6</sup> Les scripts de cette section sont inspirés du livre 24 leçons et exercices corrigés TNSI ellipses.



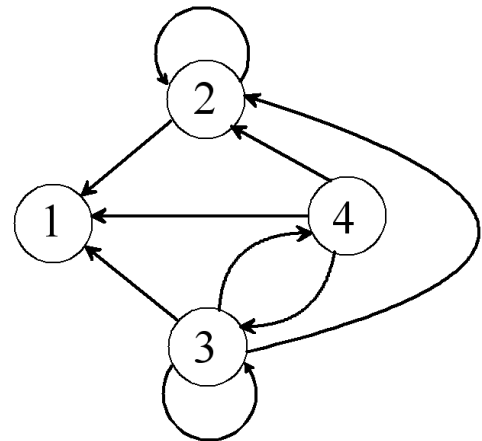
**Q5.** Cas des graphes non orientés : indiquez comment est renseignée dans ce cas la matrice d'adjacence.

**Q6.** Cas des boucles : indiquez comment est renseignée dans ce cas la matrice d'adjacence.



**Q7.** Donner la matrice d'adjacence du graphe ci-contre puis répondre aux questions :

	1	2	3	4
1				
2				
3				
4				



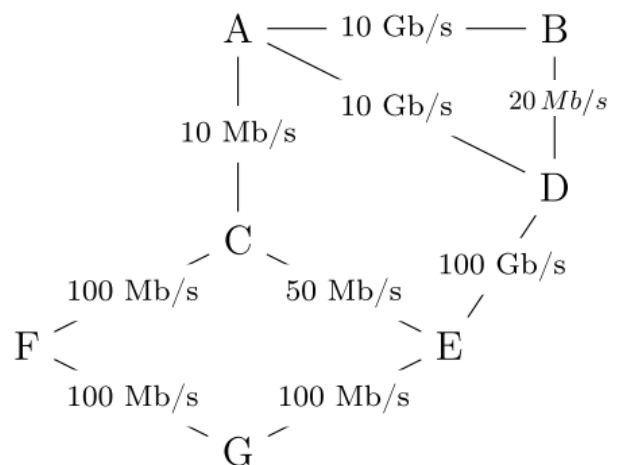
a) Comment peut-on identifier un nœud puits, c'est-à-dire un nœud d'où aucun arc ne part vers un autre nœud dans la matrice d'adjacence ?

b) Comment identifier un nœud qui est relié à tous les autres dans la matrice d'adjacence ?

### Graphes pondérés

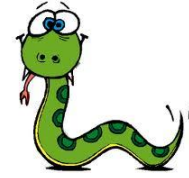
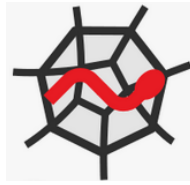
Si l'on veut indiquer un coût de transit entre deux nœuds reliés par un arc, par exemple un temps de transit, un coût en carburant, une capacité en débit d'une liaison alors on indique la valeur à coté de l'arc.

Voir l'illustration<sup>7</sup> ci-contre d'un réseau numérique interconnectant plusieurs routeurs avec les débits correspondants.

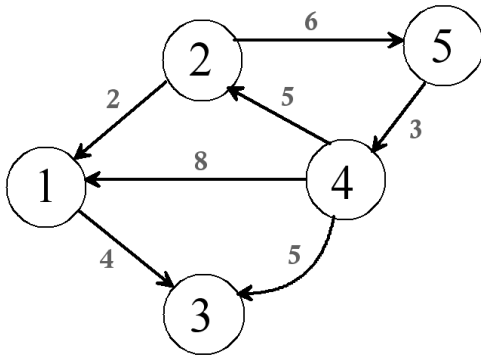


<sup>7</sup> Issue du sujet0 NSI





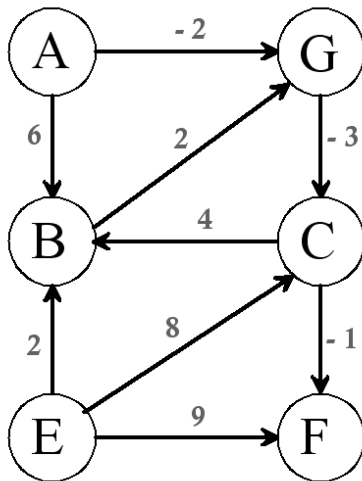
Représentation avec une matrice d'adjacence :



	1	2	3	4	5	
1		$\infty$	$\infty$	4	$\infty$	$\infty$
2		2	$\infty$	$\infty$	$\infty$	6
3		$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
4		8	5	5	$\infty$	$\infty$
5		$\infty$	$\infty$	$\infty$	3	$\infty$

Nous y voyons que l'absence de liaison entre deux nœuds à un poids d'une valeur  $\infty$ .

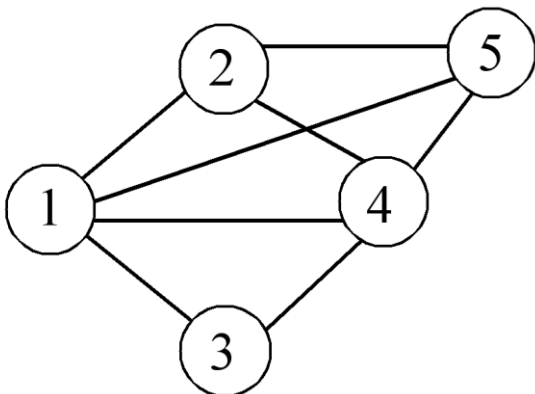
Q8. Remplir la matrice d'adjacence du graphe ci-dessous :



	A	B	C	E	F	G
A						
B						
C						
E						
F						
G						

Graphes non orientés

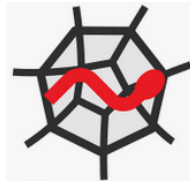
La représentation par une matrice d'adjacence de ces graphes ne pose pas de problèmes particuliers. Chaque liaison occupe deux cases dans la matrice car les liaisons sont symétriques.



	1	2	3	4	5
1		T	T	T	T
2		T	F	F	T
3		T	F	F	T
4		T	T	T	F
5		T	T	F	T

Symétrie du graphe non orienté.





La symétrie illustre le fait que les arcs ne sont pas orientés et donc que chaque liaison est bidirectionnelle.

Q9. Retracez le graphe à partir de la matrice d'adjacence ci-dessous, le graphe est-il orienté ou non orienté ?

	1	2	3	4	5
1			T	T	T
2			T		T
3	T	T			
4	T				T
5	T	T		T	

	1	2	3	4	5
1			T	T	
2	T				
3				T	
4	T				T
5		T			T

Q10. Représentez le graphe qui a la matrice d'adjacence ci-dessus.

## 2.2 Les scripts Python<sup>8</sup>

### Une classe Graphe présentation

La classe graphe ci-dessous permet de décrire les graphes avec une matrice d'adjacence. Elle est basée sur le paradigme de programmation objet.

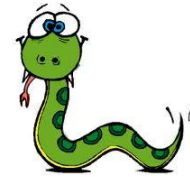
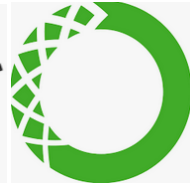
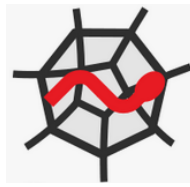
```

5 class Graphe:
6     """un graphe représenté par une matrice d'adjacence,
7       où les sommets sont les entiers 0,1,...,n-1"""
8
9     def __init__(self, n):
10        self.n = n
11        self.adj = [[False] * n for _ in range(n)]
12
13    def ajouter_arc(self, s1, s2):
14        self.adj[s1][s2] = True
15
16    def arc(self, s1, s2):
17        return self.adj[s1][s2]
18
19    def voisins(self, s):
20        v = []
21        for i in range(self.n):
22            if self.adj[s][i]:
23                v.append(i)
24        return v

```



<sup>8</sup> Les scripts sont inspirés de NSI 24 leçons avec exercices corrigés, ellipses.



Pour réviser nos connaissances en Python et POO répondre aux questions ci-dessous :

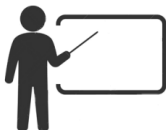
**POO**

- Q11. Quelle est l'utilité de définir une classe Graphe ?
- Q12. Comment s'appellent les fonctions décrites à l'intérieur de la classe ?
- Q13. Comment s'appellent les structures de données qui contiennent les données des entités instanciées par la Classe ?
- Q14. Quel est le nom de la fonction `__init__` ?
- Q15. Quel est son rôle ?

**Python**

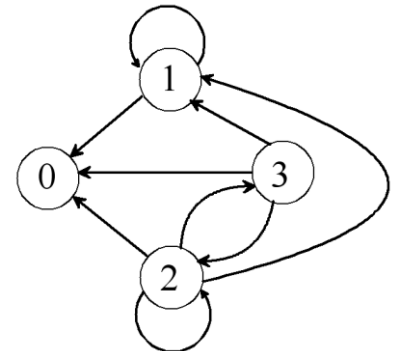
- Q16. Comment s'appelle la construction Python qui permet d'initialiser la matrice d'adjacence, ligne 11 du code ?
- Q17. Que fait l'instruction `[False]*n`

**Mise en œuvre codage d'un graphe non orienté**



Script\_Graphe\_1. Compléter le script pour décrire le graphe ci-dessous. (*Graphe identique à celui de la question Q7 mais avec les nœuds numérotés à partir de 0*)

Script\_Graphe\_1.py



Résultat attendus :

```
In [4]: g.adj
Out[4]:
[[False, False, False, False],
 [True, True, False, False],
 [True, True, True, True],
 [True, True, True, False]]
```

	0	1	2	3
0	.	.	.	.
1	T	T	.	.
2	T	T	T	T
3	T	T	T	.

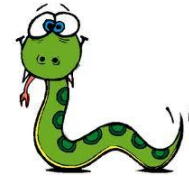
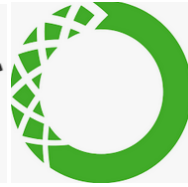
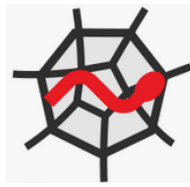
**Amélioration impression de la matrice d'adjacence**



Script\_Graphe\_2. Compléter le script 1 en ajoutant une méthode pour imprimer la matrice d'adjacence. Le format de l'impression est donné ci-dessus.

Script\_Graphe\_2.py




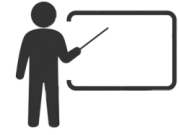


Ajout de la possibilité d'impression directement par print(objet)

Objet est un objet de la classe Graphe



Script\_Graphe\_3. Compléter le script précédent pour pouvoir imprimer la matrice d'adjacence directement en faisant `print(g)` où g est un graphe instancié par la Classe Graphe.  Script\_Graphe\_3.py



Exemple de résultat attendus :


Matrice d'adjacence

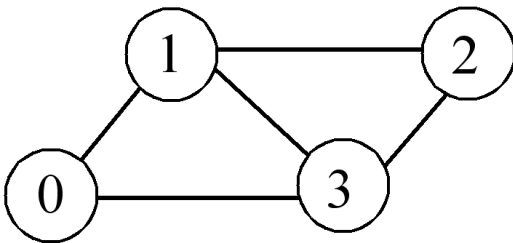
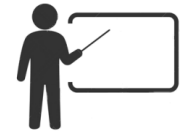
	0	1	2	3
0	.	.	.	.
1	T	T	.	.
2	T	T	T	T
3	T	T	T	.

**Q18.** Qu'est-ce qui différencient les deux fonctions d'impressions questions Q17 et Q18 ?

Ajout d'une arête (arc non orienté)



Script\_Graphe\_4. Ajouter une méthode permettant l'existence d'arêtes dans les graphes.  Script\_Graphe\_4.py



Matrice d'adjacence

	0	1	2	3
0	.	T	.	T
1	T	.	T	T
2	.	T	.	T
3	T	T	T	.

Graphe pondéré

Pour représenter cette catégorie de graphe il nous faut pouvoir utiliser la valeur infini  $\infty$  pour indiquer l'absence de liaison entre deux nœuds.

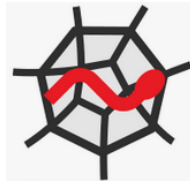
La valeur infini existe en Python :

```
In [7]: infini = float('inf')
```

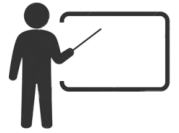
```
In [8]: infini
```

```
Out[8]: inf
```

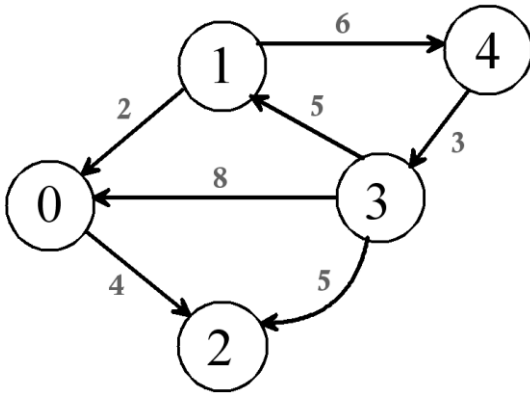




Script\_Graphe\_5. Créer une classe GraphePondere qui reprend toutes les méthodes précédemment utilisées mais adaptées à ce type de graphe. Créer le graphe ci-dessous.



Script\_Graphe\_5.py



	0	1	2	3	4
0	inf	inf	T	inf	inf
1	T	inf	inf	inf	T
2	inf	inf	inf	inf	inf
3	T	T	T	inf	inf
4	inf	inf	inf	T	inf

```
In [20]: g.arc(1,2)
Out[20]: inf
```

```
In [21]: g.arc(3,1)
Out[21]: 5
```

