Structures de données | **Chap 03** |

Les graphes

Pourquoi des graphes ?

Bien souvent, dans les problèmes que doivent résoudre les informaticien.ne.s ou tous celleux qui peuvent faire appel aux informaticien.ne.s, il existe des objets qui sont en relations les uns avec les autres.
Ces relations peuvent représenter :

— des interactions (amitiés dans un réseau social) ;

— des connexions physiques (routes, lignes de train, réseaux électriques) ;

— des liens logiques (pages web reliées par des hyperliens, dépendances entre tâches) ;

— des structures abstraites (états reliés par des transitions).

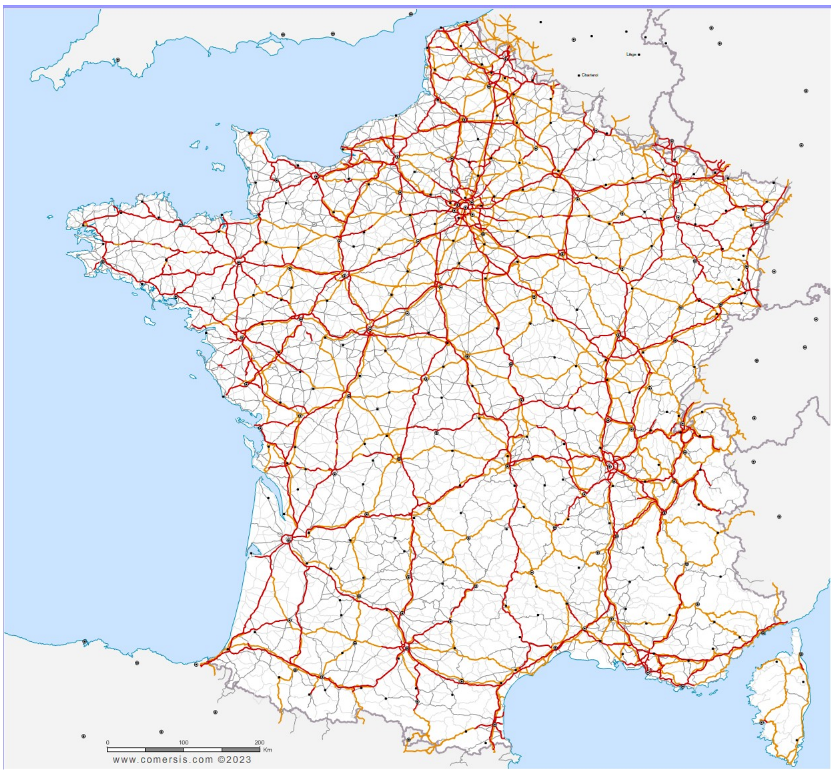
Dans toutes ces situations, ce qui est essentiel n'est pas la nature des objets eux-mêmes, mais la façon dont ils sont **reliés entre eux**. Les graphes est un concept abstrait qui permet de gérer ces situations concrètes.
La théorie des graphes permet de répondre à des questions telles que :

— Quel est la route la plus courte pour aller de Montivilliers à Marseille ?

— Est-il possible de visiter toutes les maisons d'une tournée de facteur sans passer deux fois par la même route ?

— Dans un réseau social, est-ce que deux personnes sont forcément amis d'amis d'amis... ?

— Un réseau électrique peut-il rester fonctionnel si une ligne est coupée ?



3.1 Graphes non orientés

3.1.1 Sommets et arêtes

Exercice 1 — Alliance en Terre du Milieu

On souhaite représenter les alliances au sein des personnages du Seigneur des Anneaux. Nous considérons qu'une alliance entre deux individus est réciproque, c'est-à-dire que si A est allié avec B alors B est allié avec A. Considérons les 6 personnages suivant : $Arwen, Bilbo, Celeborn, Denethor, Elrond, Faramir$.
Voici les alliances que nous considérons :

- Arwen et Elrond sont alliés
- Bilbo et Celeborn sont alliés
- Bilbo et Elrond sont alliés
- Bilbo et Denethor sont alliés
- Faramir et Arwen sont alliés
- Faramir et Bilbon sont alliés
- Faramir et Denethor sont alliés

1. Représenter la situation avec un schéma dans lequel il y a un cercle par personnage et un trait pour une alliance entre deux personnages.

2. Donner le **nombre d'alliés** que Faramir possède.

---

3. Comment Arwen pourrait-elle envoyer un messager à Celeborn sachant qu'on ne peut communiquer qu'avec quelqu'un avec qui on est allié ?

---

4. Sauron est à nos portes, Arwen décide d'envoyer un message à tous ses alliés, en leur demandant que chacun prévienne tous ses alliés respectifs, en combien d'étape tout le monde est-il prévenu ?

Définition 1 — Graphe non orienté

Un **graphe (non orienté)** est un couple $G = (S, A)$ où :

- S est un ensemble de **sommets** ;
- A est un ensemble d'**arêtes**.

Chaque arête relie deux sommets $s_1, s_2 \in S$ et est notée $\{s_1, s_2\}$. Les arêtes sont **non orientées** : $\{s_1, s_2\} = \{s_2, s_1\}$.

Définition 2 — Boucle

Dans un graphe $G = (S, A)$, une **boucle** est une arête de la forme $\{s_1, s_1\}$, c'est-à-dire une arête dont les deux extrémités correspondent au même sommet.

Définition 3 — Arêtes multiples

Dans un graphe $G = (S, A)$, on parle d'**arêtes multiples** lorsqu'il existe plusieurs arêtes distinctes ayant les mêmes extrémités, c'est-à-dire plusieurs arêtes reliant un même couple de sommets s_1 et s_2.
Un tel graphe est parfois appelé **multigraphe**.

Définition 4 — Graphe simple

Un **graphe simple** est un graphe non orienté $G = (S, A)$ tel que :

- aucune arête n'est une boucle ;
- il n'existe pas d'arêtes multiples.

Autrement dit, pour deux sommets distincts s_1 et s_2, il existe **au plus une** arête $\{s_1, s_2\}$.

Remarque 1. Dans ce chapitre, sauf mention contraire, on se place dans le cadre des graphes simples.

Exemple 1 — Le graphe de la terre du Milieu

Dans l'exercice précédent, le graphe correspond à $S = \{A, B, C, D, E, F\}$ et

$$A = \{\{A, E\}, \{B, C\}, \{B, D\}, \{B, E\}, \{A, F\}, \{B, F\}, \{F, D\}\}.$$

Définition 5 — Voisinage et sommets adjacents

Dans un graphe non orienté $G = (S, A)$, deux sommets s_1 et s_2 sont dits **adjacents** s'il existe une arête $\{s_1, s_2\} \in A$.
On dit alors que s_2 est un **voisin** de s_1 (et réciproquement).

3.1.2 Degré et poignée de mains

Définition 6 — Degré d'un sommet

Soit $G = (S, A)$ un graphe non orienté simple. Le **degré** d'un sommet $s \in S$, noté $\deg(s)$, est le nombre de sommets de S qui sont adjacents à s.
Autrement dit, $\deg(s)$ est le nombre d'arêtes de A ayant s comme extrémité.

Exercice 2 — Degrés dans les terres du Milieu

Dans l'exemple sur le Seigneur des anneaux, le degré d'un personnage est son nombre d'alliés.

1. Calculer $\deg(A), \deg(B), \deg(C), \deg(D), \deg(E), \deg(F)$.

   ----

2. Combien d'arêtes $|A|$ y a-t-il dans ce graphe ?

   ----

3. Quel sommet possède le degré maximal ?

Propriété 1 (Formule des degrés)**.** Soit $G = (S, A)$ un graphe non orienté simple. Alors :

$$\sum_{s \in S} \deg(s) = 2|A|.$$

On appelle cette propriété le lemme des poignées de main.

Idée de preuve de la formule des degrés

Chaque arête $\{s_1, s_2\} \in A$ contribue :

- pour 1 au degré de s_1 ;
- pour 1 au degré de s_2.

Ainsi, en additionnant les degrés de tous les sommets, chaque arête est comptée exactement deux fois.

Exercice 3 — Application sur le graphe de la terre du Milieu

Vérifier que notre graphe initial vérifie bien la formule des degrés.

3.1.3 Chemins, cycles et connexité

Définition 7 — Chemin

Soit $G = (S, A)$ un graphe non orienté. Un **chemin** de longueur $k \geq 1$ est une suite de sommets

$$(s_0, s_1, \ldots, s_k)$$

telle que pour tout $i \in \{0, \ldots, k-1\}$, on ait $\{s_i, s_{i+1}\} \in A$.

Définition 8 — Cycle

Un **cycle** est un chemin $(s_0, \ldots, s_k)$ avec $k \geq 1$ tel que $s_0 = s_k$.

Définition 9 — Graphe connexe

Un graphe non orienté $G = (S, A)$ est **connexe** si pour toute paire de sommets $s_1, s_2 \in S$, il existe un chemin allant de s_1 à s_2.

Exercice 4 — Connexité et chemins

On reprend le graphe de la Terre du Milieu

1. Le graphe est-il connexe ? Justifier en utilisant la définition.

   ....

   ....

   ....

2. Donner un chemin de C vers A.

   ....

   ....

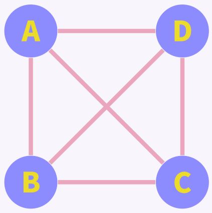
3. Donner un cycle (s'il en existe un). Sinon, expliquer pourquoi.

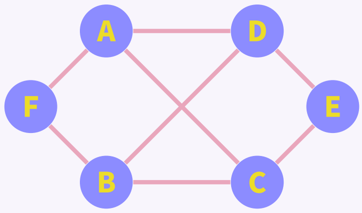
   ....

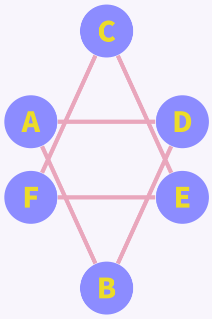
   ....

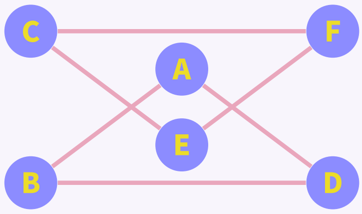
Exercice 5 — Connexité

Pour chaque graphe ci-dessous, indiquer s'il est **connexe** ou **non connexe**.









3.2 Chemins et cycles eulériens

Exercice 6 — A la découverte des chemins eulériens

1. Redessiner le graphe de la terre du milieu en dessous.

2. Est-il possible de trouver un chemin dans le graphe de la Terre du Milieu qui passe exactement une seule fois par chaque arête ? Si oui, lequel ?

----

3. Même question mais avec un cycle qui passe exactement une seule fois par chaque arête.

Définition 10 — Chemin eulérien et cycle eulérien

Dans un graphe non orienté $G = (S, A)$:

— un **chemin eulérien** est un chemin qui passe **exactement une fois** par chaque arête ;

— un **cycle eulérien** est un chemin eulérien qui est en plus un **cycle**.

Propriété 2 (Critère d'Euler)**.** Soit $G = (S, A)$ un graphe non orienté **connexe**.

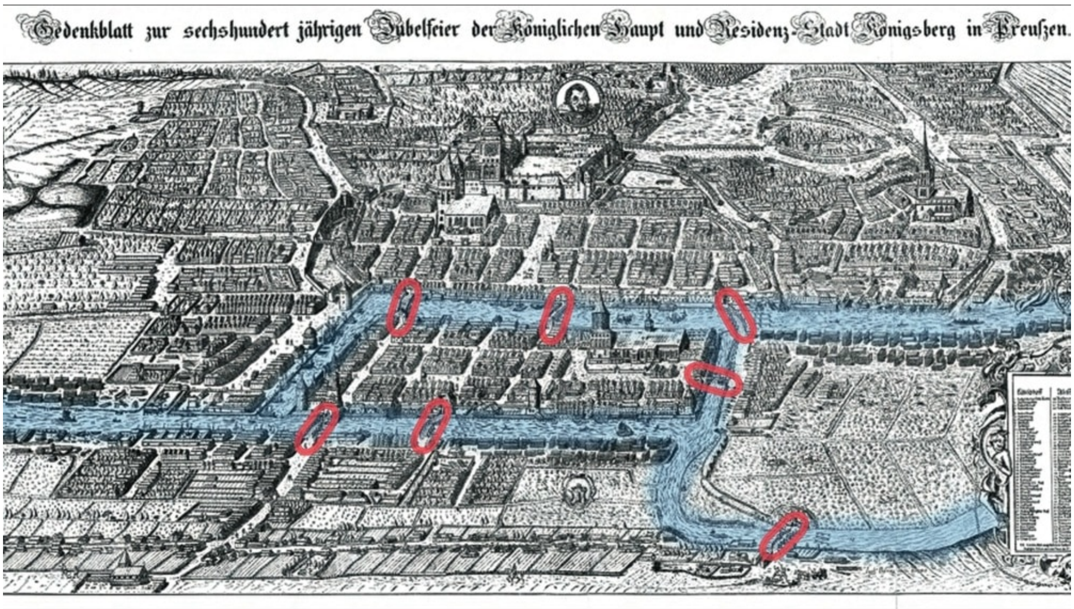
— G possède un **chemin eulérien** (non fermé) si et seulement s'il possède exactement **deux** sommets de degré impair.

— G possède un **cycle eulérien** si et seulement si tous ses sommets sont de degré **pair**.

3.2.1 Les 7 ponts de Königsberg

Le problème de Königsberg

La ville de Königsberg avait 7 ponts qui traversent une rivière telle qu'illustrée ci-après. Euler se demanda s'il était possible de faire une promenade en traversant tous les ponts une seule fois et revenir à son point de départ. En représentant le problème par un graphe, on peut alors reformuler la question de la promenade d'Euler ainsi : peut-on parcourir toutes les arêtes du graphe une unique fois et revenir à notre point de départ ?



Exercice 7 — Modéliser et conclure

On modélise la ville par un graphe :

— chaque zone de terre est un sommet ;

— chaque pont est une arête.

1. Proposer un graphe (schéma) représentant les 7 ponts.

2. Calculer le degré de chaque sommet.

3. Le graphe est-il connexe ?

4. En utilisant le critère d'Euler, conclure : existe-t-il un cycle eulérien ?

5. Proposer une modification pour rendre l'existence d'un cycle eulérien possible.

3.2.2 Application

Exercice 8 — Chaînes et cycles eulériens — application

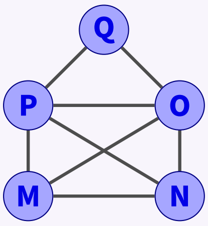
Pour chacun des graphes ci-dessous :

— déterminer le degré de chaque sommet ;

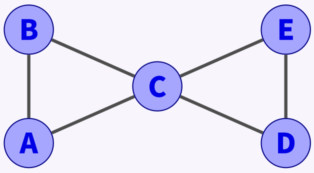
— dire s'il existe un chemin eulérien

— dire s'il existe un **cycle eulérien** ;

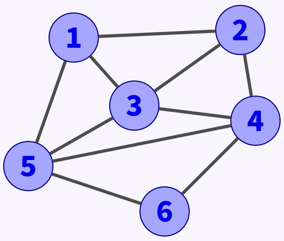
— justifier à l'aide du **critère d'Euler**.



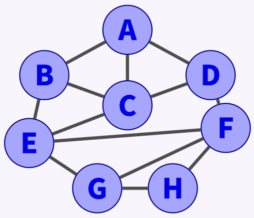
Graphe 1



Graphe 2



Graphe 3



Graphe 4

1. Graphe 1 :

   ----

2. Graphe 2 :

   ----

3. Graphe 3 :

   ----

4. Graphe 4 :

3.3 Graphes orientés

Exercice 9 — Allégeances dans la Terre du Milieu

On souhaite modéliser des **allégeances** dans la Terre du Milieu. Contrairement à une alliance, une allégeance **n'est pas réciproque** : si s_1 prête allégeance à s_2, cela ne signifie pas que s_2 prête allégeance à s_1.

On considère les personnages suivants : $S = \{Aragorn, Théoden, Denetho, Boromir, Faramir, Legolas, G$

Les allégeances suivantes sont établies :

— Boromir prête allégeance à Denethor ;

— Aragorn prête allégeance à Boromir ;

— Boromir prête allégeance à Aragorn

— Faramir prête allégeance à Denethor ;

— Aragorn prête allégeance à Théoden

— Théoden prête allégeance à Aragorn ;

— Legolas prête allégeance à Aragorn ;

— Gimli prête allégeance à Aragorn ;

— Sauron ne prête allégeance à personne ;

— Nazgûl prête allégeance à Sauron.

1. Représenter la situation par un schéma (un cercle par personnage, et une flèche pour chaque allégeance).

2. Pour chaque personnage, donner :

— le nombre d'allégeances qu'il prête

— le nombre d'allégeances qu'il reçoit.

----

3. Est-ce qu'on peut créer une chaîne d'allégeance de n'importe quel personnage à n'importe quel personnage ?

Définition 11 — Graphe orienté

Un **graphe orienté** est un couple $G = (S, A)$ où :

- S est un ensemble de sommets ;
- A est un ensemble d'**arcs**.

Chaque arc est un couple ordonné (s_1, s_2) de sommets de S, ce qui signifie que l'arc part de s_1 et arrive en s_2.

Définition 12 — Sommet prédécesseur et successeur

Dans un graphe orienté $G = (S, A)$:

- on dit que s_1 est un **prédécesseur** de s_2 s'il existe un arc $(s_1, s_2) \in A$;
- on dit que s_2 est un **successeur** de s_1.

Définition 13 — Degré entrant et degré sortant

Dans un graphe orienté $G = (S, A)$:

- le **degré entrant** d'un sommet s, noté $\deg^-(s)$, est le nombre d'arcs arrivant en s ;
- le **degré sortant** d'un sommet s, noté $\deg^+(s)$, est le nombre d'arcs partant de s.

Remarque 2. Dans un graphe orienté, le degré total d'un sommet n'a pas de sens unique : il faut toujours distinguer le degré entrant et le degré sortant.

Définition 14 — Chemin orienté

Soit $G = (S, A)$ un graphe orienté. Un **chemin orienté** est une suite de sommets

$$(s_0, s_1, \ldots, s_k)$$

telle que, pour tout $i \in \{0, \ldots, k-1\}$, $(s_i, s_{i+1}) \in A$.
Le sens des arcs doit donc être respecté tout au long du chemin.

Définition 15 — Graphe orienté fortement connexe

Un graphe orienté $G = (S, A)$ est dit **fortement connexe** si, pour toute paire de sommets $s_1, s_2 \in S$, il existe :

— un chemin orienté de s_1 vers s_2 ;

— et un chemin orienté de s_2 vers s_1.

3.4 Graphes pondérés

Exercice 10 — Voyager dans la Terre du Milieu

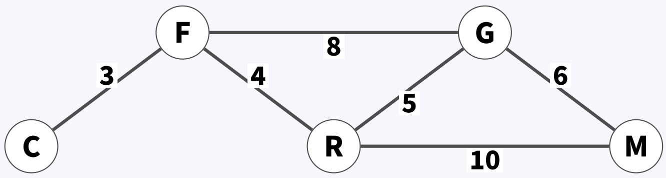
On souhaite modéliser les déplacements entre plusieurs lieux importants de la Terre du Milieu. Chaque lieu est représenté par un sommet, et chaque route par une arête.

Contrairement aux exemples précédents, toutes les routes n'ont pas la même difficulté : certaines sont longues, dangereuses ou traversent des régions hostiles. On associe donc un **poids** à chaque arête.

Les sommets représentent les lieux suivants :

$$S = \{\text{Fondcombe } (F),\ \text{Rohan } (R),\ \text{Gondor } (G),\ \text{Mordor } (M),\ \text{Comté } (C)\}.$$

Les routes et leurs difficultés (poids) sont données ci-dessous :



Le poids d'une arête représente la **difficulté du trajet** (distance, danger, fatigue, etc.).

1. Quel est le poids du chemin :

$$C \to F \to R \to G \to M?$$

----

2. Quel est le poids du chemin :

$$C \to F \to G \to M?$$

----

3. Quel est le poids du chemin :

$$C \to F \to R \to M?$$

----

4. Parmi ces trois chemins, lequel est le **moins coûteux** ?

----

5. Peut-on atteindre Mordor depuis la Comté sans passer par Fondcombe ? Justifier.

Définition 16 — Graphe pondéré (non orienté)

Un **graphe pondéré** est un graphe non orienté $G = (S, A)$ auquel est associée une fonction de poids

$$w : A \to \mathbb{R} \quad (\text{souvent } \mathbb{N}).$$

À chaque arête $\{s_1, s_2\} \in A$ est associé un nombre $w(\{s_1, s_2\})$, appelé **poids** de l'arête.

Exemple 2 — Interprétation d'un poids

Selon le contexte, le poids d'une arête peut représenter :

— une distance (en kilomètres) ;

— un temps de trajet (en minutes) ;

— un coût financier ;

— une difficulté ou un danger.

Le graphe est le même, seule l'interprétation du poids change.

3.4.1 Poids d'un chemin

Définition 17 — Poids d'un chemin

Soit $G = (S, A)$ un graphe pondéré et

$$(s_0, s_1, \dots, s_k)$$

un chemin dans G.

Le **poids** (ou **coût**) de ce chemin est la somme des poids des arêtes empruntées :

$$w(s_0, \dots, s_k) = \sum_{i=0}^{k-1} w(\{s_i, s_{i+1}\}).$$

Remarque 3. Deux chemins reliant les mêmes sommets peuvent avoir des poids différents. Le chemin le plus court en nombre d'arêtes n'est pas forcément le moins coûteux.

3.5 Représentations des graphes

Pour l'instant, nous avons représenté les graphes uniquement grâce à des schémas mais cela n'est pas évident d'implémenter aisément un schéma dans un langage de programmation. Pour cela, nous utilisons d'autres représentations, la première est la matrice d'adjacence.

3.5.1 Par Matrice d'Adjacence

Définition 18 — Matrice d'adjacence (graphe non orienté)

Soit $G = (S, A)$ un graphe non orienté, et supposons $S = \{s_1, \dots, s_n\}$. La **matrice d'adjacence** de G est la matrice M remplie uniquement avec des 1 et 0 et définie par :

$$M_{i,j} = \begin{cases} 1 & \text{si } \{s_i, s_j\} \in A, \\ 0 & \text{sinon.} \end{cases}$$

Autrement dit, il y a une arête en case (i, j) si il y a une arête entre s_i et s_j. Dans un graphe non orienté, M est **symétrique** : $M_{i,j} = M_{j,i}$.

Exemple 3 — Matrice d'adjacence

Prenons l'exemple du graphe de la Terre du Milieu, voici sa matrice d'adjacence.

	A	B	C	D	E	F
A	0	0	0	0	1	1
B	0	0	1	1	1	1
C	0	1	0	0	0	0
D	0	1	0	0	0	1
E	1	1	0	0	0	0
F	1	1	0	1	0	0

Exercice 11 — Construire une matrice d'adjacence

Pour chacun des 4 graphes de l'exercice 5, écrire les matrices d'adjacence.

Lire le degré dans la matrice

Dans un graphe non orienté **sans boucle**, le degré d'un sommet s_i est la somme des éléments de la ligne i :

$$\deg(s_i) = \sum_{j=1}^{n} M_{i,j}.$$

3.5.2 Par liste d'adjacence

Une autre manière de faire est de garder en mémoire les voisins de chaque sommet.

Définition 19 — Liste de voisins (graphe non orienté)

Soit $G = (S, A)$ un graphe non orienté, et supposons $S = \{s_1, \dots, s_n\}$. La **liste de voisins** (ou **liste d'adjacence**) de G est la représentation qui associe à chaque sommet $s \in S$ l'ensemble (ou la liste) de ses voisins :

$$\text{Voisins}(s) = \{\, t \in S \mid \{s, t\} \in A \,\}.$$

Remarque 4. Dans un graphe non orienté, si $t \in \text{Voisins}(s)$ alors $s \in \text{Voisins}(t)$.

Exemple 4 — Liste de voisins

On reprend le graphe de la Terre du Milieu Sa liste de voisins est :

$$\begin{aligned}
A &: \{E, F\} \\
B &: \{C, D, E, F\} \\
C &: \{B\} \\
D &: \{B, F\} \\
E &: \{A, B\} \\
F &: \{A, B, D\}
\end{aligned}$$

Lien avec la matrice d'adjacence

Pour un sommet s_i, la liste de voisins correspond aux colonnes j telles que $M_{i,j} = 1$. Autrement dit, $\text{Voisins}(s_i) = \{s_j \mid M_{i,j} = 1\}$.

Exercice 12 — Construire une liste de voisins

Pour chacun des 4 graphes de l'exercice 8 :

1. écrire la liste de voisins de chaque sommet ;
2. vérifier que la réciprocité est respectée (si X est voisin de Y, alors Y est voisin de X).

Matrice ou liste de voisins ?

Les deux représentations contiennent la même information, mais ne sont pas toujours pratiques dans les mêmes cas :

- La **matrice d'adjacence** permet de savoir très vite si deux sommets sont adjacents (il suffit de regarder $M_{i,j}$), mais elle prend beaucoup de place si le graphe a peu d'arêtes.
- La **liste de voisins** est souvent plus compacte lorsque le graphe contient peu d'arêtes : on ne stocke que les voisins réellement existants.