



► Compétences

- ✓ Maîtriser le vocabulaire des graphes (sommets, arcs, chemins, circuits)
- ✓ Représenter une situation à l'aide d'un graphe orienté
- ✓ Utiliser les puissances de la matrice d'adjacence pour dénombrer des chemins
- ✓ Déterminer et interpréter la fermeture transitive d'un graphe



► Fil rouge *coloration drone*

Un essaim de drones communique via des liaisons radio *point-à-point*, dont certaines sont à sens unique selon la portée et le relief survolé.

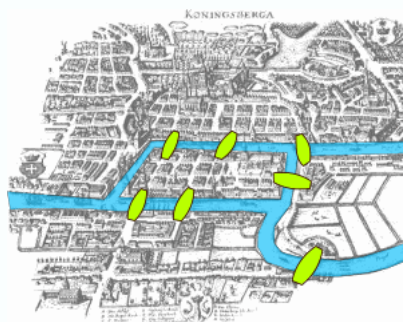
Comment modéliser ce réseau de communication, et vérifier que chaque drone peut, directement ou indirectement, transmettre une alerte à n'importe quel autre drone de l'essaim ?



I. Exemples

► Exemple

On peut commencer par le problème des « sept ponts de Königsberg » (exemple historique lié à Euler) :



La ville de Königsberg (aujourd'hui Kaliningrad) est construite autour de deux îles situées sur le Pregel et reliées entre elles par un pont. Six autres ponts relient les rives de la rivière à l'une ou l'autre des deux îles, comme représentés sur le plan ci-dessus. Le problème consiste à déterminer s'il existe ou non une promenade dans les rues de Königsberg permettant, à partir d'un point de départ au choix, de passer une et une seule fois par chaque pont, et de revenir à son point de départ, étant entendu qu'on ne peut traverser le Pregel qu'en passant sur les ponts.

► Point histoire (*Anecdote*)



Leonhard Euler (1707 – 1783, ) est considéré par certains comme le *plus grand* mathématicien de tous les temps. Terriblement prolifique, il publie plus de 800 pages par an!! Doté d'une mémoire fabuleuse, il calcula une nuit d'insomnie les puissances 6^{ème} de tous les entiers de 1 à 100 et s'en souvint plusieurs jours plus tard.

Une fièvre brutale lui fit perdre son œil droit.

À la cour de Berlin, Frédéric le Grand, préférant les esprits brillants comme Voltaire aux scientifiques efficaces, le traite de « cyclope mathématique ». Euler a résolu ce problème des « sept ponts de Königsberg » en 1735.

► Exemple 1.1

Dans une ville, on considère 4 carrefours A, B, C, D reliés par des rues à sens unique ou à double sens. On a les renseignements suivants :

- une rue à sens unique va de A à B ;
- une rue à sens unique va de A à C ;
- une rue à sens unique va de C à D ;
- une rue à sens unique va de D à D sans passer par A, B ou C ;
- une rue en double sens relie A et D ;
- une rue en double sens relie B et C.

On peut représenter cette situation par :

- un graphique ;
- un tableau des successeurs ;
- un tableau de prédécesseurs ;
- un tableau à double entrée ;
- une matrice d'adjacence.

II. Définitions, vocabulaire

II.1. Graphe

► Définitions 2.1

Un **graphe** est un ensemble de points appelé sommets et de liaisons appelées arcs. (Dans notre étude, ces arcs seront simples et orientés. Ils pourront être pondérés)

Une **boucle** est un arc dont les extrémités sont confondues

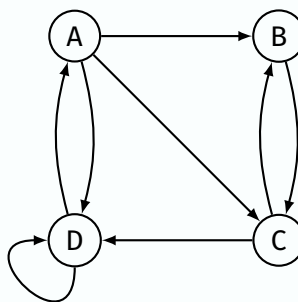
Un **chemin** est une suite d'arcs tels que chaque fin d'arc est le début de l'arc suivant (s'il y en a un).

Un **circuit** est un chemin dont le premier et le dernier sommet sont identiques.

Un **chemin hamiltonien** est un chemin qui passe une fois et une seule par chacun des sommets du graphe.

► Exemple 2.1

L'exemple introductif peut-être représenté par le graphe $\mathcal{G}$:



II.2. Matrice d'adjacence

► Définition 2.2

Soit $\mathcal{G}$ un graphe dont les sommets sont $A_1, A_2, \dots, A_n$

La matrice d'adjacence du graphe $\mathcal{G}$ est la matrice carrée d'ordre n , notée $M = (a_{ij})$ telle que

$$a_{ij} = \begin{cases} 1 & \text{si } A_i - A_j \text{ est un arc du graphe} \\ 0 & \text{sinon} \end{cases}.$$

► Exemple 2.2

La matrice M du graphe de l'exemple introductif est :

$$M = \begin{matrix} & \begin{matrix} A & B & C & D \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

► Remarques 2.1

Pour interpréter plus facilement les matrices, il est commode de « border » la matrice avec les sommets.

Il faut pas oublier le « sens de lecture » de la matrice d'adjacence :

- les arcs existants vont des lignes vers les colonnes ;
- une ligne donne les successeurs ;
- une colonne donne les prédécesseurs ;
- la diagonale donne les boucles.

III. Applications

III.1. Longueur d'un chemin

► Définition 3.1

La **longueur** d'un chemin est le nombre d'arcs qui le constituent.

Un chemin de n sommets est de longueur $n - 1$.

► Propriété 3.1

Si M est la matrice d'adjacence d'un graphe simple orienté de sommets $A_1, A_2, \dots, A_n$, le nombre de chemins de longueur p d'un sommet A_i à un sommet A_j est le nombre situé ligne i et colonne j dans la matrice M^p .

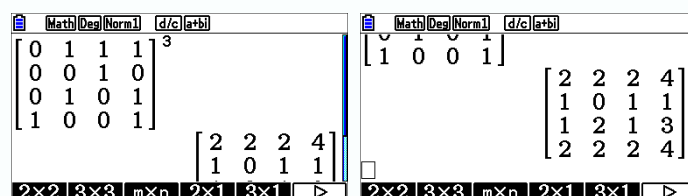
► Exemple 3.1

Pour le graphe donné en exemple introductif, on a, par exemple,

$$M^3 = \begin{matrix} & \begin{matrix} A & B & C & D \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \end{matrix} & \begin{pmatrix} 2 & 2 & 2 & 4 \\ 1 & 0 & 1 & 1 \\ 1 & 2 & 1 & 3 \\ 2 & 2 & 2 & 4 \end{pmatrix} \end{matrix}$$

De ce fait, on peut affirmer qu'il y a 2 chemins de longueur 3 allant de C à B :

- C – B – C – B ;
- C – D – A – B.



► Python

En , on peut – grâce au module  – calculer le nombre de chemins de longueur donnée :

</> Code Python

```
1 import numpy as np
2
3 def nb_chemins(coeffs, puiss, deb, fin) :
4     M=np.asmatrix(coeffs)**puiss
5     res = M.item((deb-1, fin-1))
6     print(M)
7     print(f"Il y a {res} chemin(s) de lg {puiss} allant du som num {deb} au som num {fin}")
```

♣ Début de la console Python ♣

```
>>> nb_chemins('[0 1 1 1 ; 0 0 1 0 ; 0 1 0 1 ; 1 0 0 1]', 3, 3, 2)
[[2 2 2 4]
 [1 0 1 1]
 [1 2 1 3]
 [2 2 2 4]]
Il y a 2 chemin(s) de longueur 3 allant du sommet num 3 au sommet num 2
```

♣ Fin de la console Python ♣

► Définition 3.2

On note $M^{[n]}$ la n -ième puissance booléenne de la matrice M .

Elle se calcule comme M^n en utilisant l'addition et la multiplication booléenne.

► Propriété 3.2

Il existe au-moins un chemin de longueur p du sommet A_i au sommet A_j ssi le coefficient (ij) de la matrice $M^{[p]}$ est égale à 1.

III.2. Fermeture transitive

► Définition 3.3

La fermeture transitive du graphe $\mathcal{G}$ est le graphe $\widehat{\mathcal{G}}$ constitué des sommets et des arcs de $\mathcal{G}$ auxquels on ajoute, si nécessaire, les arcs $(A ; B)$ pour lesquels il existe un chemin (de longueur quelconque) allant du sommet A au sommet B .

► Théorème 3.1

La matrice $\widehat{M}$ de la fermeture transitive d'un graphe $\mathcal{G}$ à n sommets est donnée par :

$$\widehat{M} = M \oplus M^{[2]} \oplus M^{[3]} \oplus \dots \oplus M^{[n]}$$

avec $\oplus$ l'addition booléenne et $[i]$ la puissance booléenne.

On dit que $\widehat{M}$ est la matrice d'accessibilité du graphe $\mathcal{G}$

► Exemple 3.2

Pour le graphe donné en exemple introductif, on a

$$M + M^2 + M^3 + M^4 = \begin{matrix} & \begin{matrix} A & B & C & D \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \end{matrix} & \begin{pmatrix} 7 & 8 & 8 & 15 \\ 2 & 3 & 3 & 5 \\ 5 & 5 & 5 & 10 \\ 8 & 7 & 7 & 15 \end{pmatrix} \end{matrix}.$$

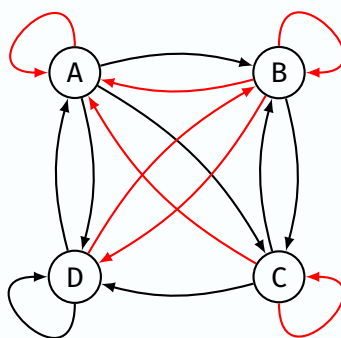
De ce fait, on obtient $\widehat{M} = \begin{matrix} & \begin{matrix} A & B & C & D \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \end{matrix} & \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \end{matrix}.$

Concrètement, de chacun des sommets, on peut aller à tous les autres.

	1	2	3	4
1	0	1	1	1
2	0	0	1	0
3	0	1	0	1
4	1	0	0	1

	1	2	3	4
1	7	8	8	15
2	2	3	3	5
3	5	5	5	10
4	8	7	7	15

Pour construire le graphe $\widehat{\mathcal{G}}$, on complète le graphe $\mathcal{G}$ avec les **nouvelles arêtes venant des 1 de $\widehat{M}$** :



► Python

En , on peut – grâce au module  – calculer (sans la passer en booléen...) la :

</> Code Python

```
1 def fermeture_transitive(coeffs) :
2     M=np.asmatrix(coeffs)
3     nb = M.shape[1]
4     for i in range(2, nb+1) :
5         M += np.mat(coeffs)**i
6     return M
```



♣ Début de la console Python ♣

```
>>> FT = fermeture_transitive('[0 1 1 1 ; 0 0 1 0 ; 0 1 0 1 ; 1 0 0 1]')
>>> FT
matrix([[ 7,  8,  8, 15],
        [ 2,  3,  3,  5],
        [ 5,  5,  5, 10],
        [ 8,  7,  7, 15]])
```

♣ Fin de la console Python ♣

III.3. Circuits

► Idées

Le fait qu'un graphe ne possède pas de circuit va permettre dans une certaine mesure de le *hierarchiser*.

La recherche de l'existence ou non de circuits dans un graphe peut être fastidieuse, et sort du cadre du programme de BTS SIO.

► Théorème 3.2

Le graphe $\mathcal{G}$ est sans circuit si et seulement si sa fermeture transitive $\widehat{\mathcal{G}}$ n'a pas de boucle, c'est-à-dire si et seulement si la diagonale de $\widehat{M}$ ne contient que des 0.

IV. Compléments

IV.1. Niveau d'un sommet

► Définition 4.1 (Méthode)

Dans un graphe sans circuit, le niveau d'un sommet X est la longueur du plus long chemin d'extrémité X .

On peut représenter un graphe ordonné par niveaux, avec comme effet de n'avoir aucun arc entre deux sommets d'un même niveau (autrement dit aucun arc « vertical ») !

► Méthode 4.1

Pour déterminer le niveau des sommets d'un graphe sans circuit :

- N_0 est constitué des sommets sans prédécesseur ;
- on « barre » les sommets de N_0 ;
- N_1 est constitué des (nouveaux) sommets sans prédécesseur ;
- on « barre » les sommets de N_1 ;
- N_2 est constitué des (nouveaux) sommets sans prédécesseur ;
- etc

On peut ensuite représenter le graphe, de manière « *horizontale* », de sorte que les arcs iront **toujours** de la gauche vers la droite :

 N_0 N_1 N_2

etc

IV.2. Graphes valués

► Définition 4.2

Un graphe **valué** (ou **pondéré**) est un graphe pour lequel on a affecté une valeur à chacun des arcs.

► Propriété 4.1

La **valeur** d'un chemin est égale à la somme des valeurs des arcs de ce chemin

► Définition et propriété 4.3

Un chemin **optimal** est un chemin de valeur (ou poids) minimale ou maximale d'une sommet à l'autre.

On peut utiliser l'algorithme de **Dijkstra** ou le **marquage de Ford** pour déterminer une plus courte chaîne entre deux sommets.

IV.3. Arborescence

► Définition 4.4

Une arborescence est un graphe qui possède un sommet et un seul nommé **racine** tel que tout sommet peut être atteint par un chemin et un seul issu de cette racine.

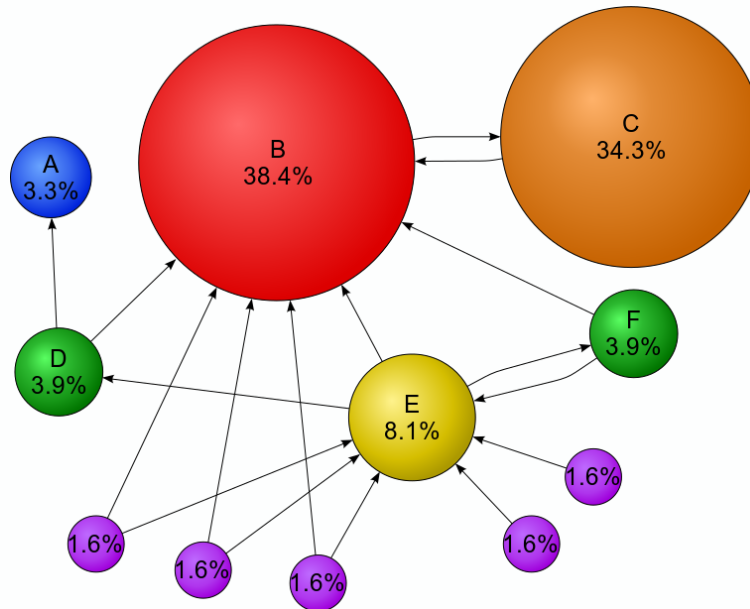
On peut utiliser une arborescence pour : des arbres généalogiques, des arbres de probas, hiérarchiques, etc

IV.4. Le Pagerank

► Idée

Le web peut être vu comme un graphe (orienté) où les sommets sont les pages web et les arcs sont les hyperliens entre les pages.

L'intuition qui se cache derrière l'algorithme *PageRank*, de Google, est que la présence d'un lien vers une certaine page constitue un « vote » en faveur de cette page. L'algorithme *PageRank* utilise donc des matrices carrées *gigantesques* pour « prédire » la popularité d'une page web.



► Point histoire



L'idée de PageRank est officiellement présentée pour la première fois en 1998 par Sergey Brin et Larry Page, les fondateurs de Google, dans « The Anatomy of a Large-Scale Hypertextual Web Search Engine ».

Le premier brevet, *Method for Node Ranking in a Linked Database*, est cependant déposé dès janvier 1997 avant d'être enregistré le 9 janvier 1998. Il est d'abord la propriété de l'Université Stanford, qui octroie ensuite la licence à Google la même année (amendée en 2000 et 2003), deux mois après sa fondation.