Partie1-Graphes et algorithmes

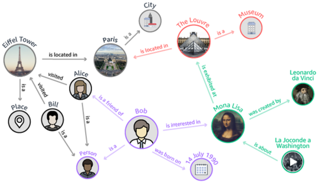
FSM, Université de Monastir

October 30, 2025

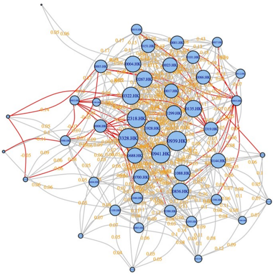
Chapitre1: Généralités

Motivation

(a) Réseaux sociaux

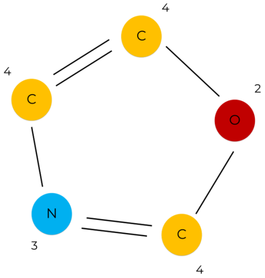


(b) Graphe de connaissances

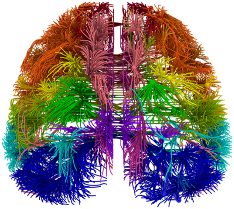


(c) Finance

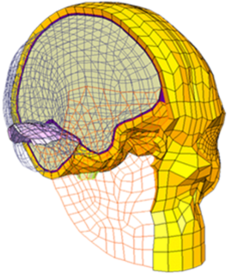
Motivation



(a) Chimie



(b) Santé

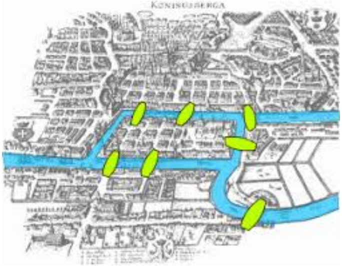


(c) Images volumétriques

Historique

Problème des sept ponts de Königsberg

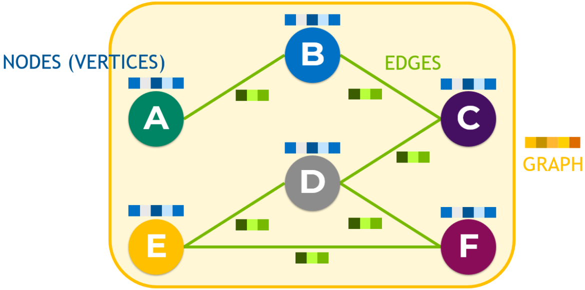
A priori le premier problème résolu par la théorie des graphes (en 1736). Euler démontra qu'il était impossible de traverser chacun des sept ponts de la ville russe de Konigsberg une fois exactement et de revenir au point de depart. Les ponts enjambent les bras de la Pregel qui coulent de part et d'autre de l'île de Kneiphof.



Graphe

Un graphe est défini par un couple $G = (V, E)$ tel que:

- V est un ensemble fini de n sommets(Vertices en anglais),
- E est un ensemble de m couples de sommets$(v_i, v_j) \in V^2$, appelés arêtes (Edges en anglais).



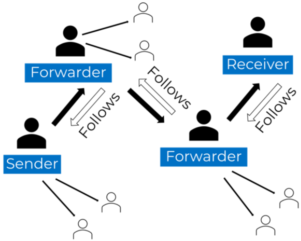
Edges

Un graphe peut être :

- **orienté**
- **non orienté**
- **pondéré**
- **non pondéré**

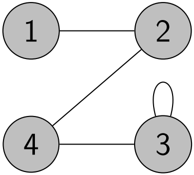
FACEBOOK

TWITTER



Graphe non orienté

Dans un graphe non orienté, les couples $(v_i, v_j) \in V$ ne sont pas orientés, c'est à dire que (v_i, v_j) est équivalent à (v_j, v_i). Une paire (v_i, v_j) est appelée une arête, et est il représentée graphiquement par $v_i - v_j$.



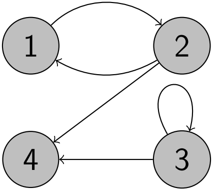
$G = (V, E)$ avec $V = \{1, 2, 3, 4\}$,$E = \{(1, 2), (2, 4), (3, 4), (3, 3)\}$

Remarque

Une **boucle** est une arrête reliant un sommet à lui même.

Graphe orienté

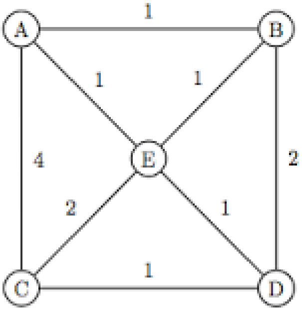
Dans un graphe orienté, les couples$(s_i, s_j) \in V$ sont orientés, c'est à dire que (s_i, s_j) est un couple ordonné, où s_i est le sommet initial, et s_j le sommet terminal. Un couple(s_i, s_j) est appelé un **arc**, et il est représenté graphiquement par $s_i \rightarrow s_j$.



$G = (V, E)$ avec $V = \{1, 2, 3, 4\}$,
$E = \{(1, 2), (2, 1), (2, 4), (3, 4), (3, 3)\}$

Graphe pondéré

Un graphe pondéré est un graphe étiqueté où chaque arrête/arc est affectée d'un nombre réel , appelé poids de cette arrête/arc.



Terminologie

Dans un graphe G on appelle:

- **Ordre** d'un graphe le nombre de sommets n de ce graphe.
- **Taille** d'un graphe est le nombre d'arêtes m ou d'arcs.

Degré d'un sommet-graphe non orienté

Dans un graphe non orienté, on appelle degré du sommet v, et on note $d(v)$, le nombre d'arêtes incidentes à ce sommet.
Attention ! Une boucle sur un sommet compte double.

Theorem

La somme des degrés des sommets d'un graphe non orienté simple est égale à deux fois le nombre d'arêtes.

Terminologie

Theorem

*Dans un **graphe simple**, il y a un nombre pair de sommets de degré impair.*

Degré d'un sommet-graphe orienté

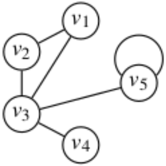
- Dans un graphe orienté, le **demi-degré extérieur** ou **demi-degré sortant** d'un sommet s, noté $d^+(s)$, est le nombre d'arcs partant de s, i.e. de la forme (s, v) .
- De même, le **demi-degré intérieur** ou **demi-degré entrant** d'un sommet s, noté $d^-(s)$, est le nombre d'arcs arrivant en s, i.e. de la forme (v, s) .
- Le degré d'un sommet s est alors la somme des degrés entrant et sortant : $d(s) = d^+(s) + d^-(s)$

Notions élémentaires

Degré d'un graphe

Le degré d'un graphe est le **degré maximum** de tous ses sommets.

Dans l'exemple ci-dessous, le degré du graphe est 4, à cause du sommet v_3 .



Un graphe est dit régulier si tous les sommets ont le même degré. Si le degré commun est k , alors on dit que le graphe est k-**régulier**.

Notions élémentaires

- Deux sommets sont **adjacents, ou incidents** s'il existe une arête e qui relie ces sommets.
- Une arête e est **incidente** à un sommet v_i si v_i est une extrémité de e.
- Deux arêtes incidentes à un même sommet sont **adjacentes**.
- Un sommet est **isolé** s'il n'existe pas d'arête incidente à ce sommet.

Densité d'un graphe

On peut associer à tout graphe un entier appelé densité du graphe.

Definition

La densité d'un graphe $G = (V; E)$ est définie par le rapport entre le nombre d'arêtes divisé par le nombre d'arêtes possibles:

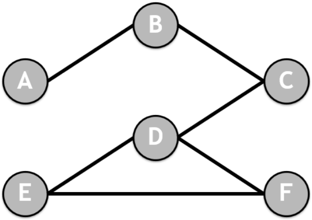
$$D = \frac{2 \times |E|}{|V| \times (|V| - 1)}$$

- Ce paramètre mesure si le graphe a beaucoup d'arêtes ou peu. Un graphe **dense** (dense graph) est un graphe dans lequel le nombre d'arêtes est proche du nombre maximal.
- Un graphe **creux** (sparse graph) a au contraire peu d'arêtes
- La densité 0 correspond au graphe où tous les sommets sont **isolés**, et la densité 1 au graphe **complet**.

Matrice d'adjacence

La matrice d'adjacence est une matrice en deux dimensions. Chacune des dimensions est indexée par les sommets du graphe. A l'intersection de chaque ligne et colonne on trouve un nombre: il vaut 1 si une arête relie les deux sommets indexés par les coordonnées de la case, et 0 sinon.

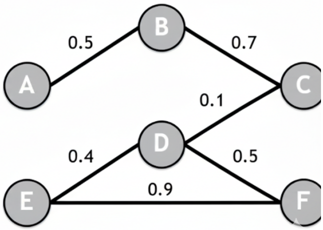
	A	B	C	D	E	F
A	0	1	0	0	0	0
B	1	0	1	0	0	0
C	0	1	0	1	0	0
D	0	0	1	0	1	1
E	0	0	0	1	0	1
F	0	0	0	1	1	0



Matrice d'adjacence

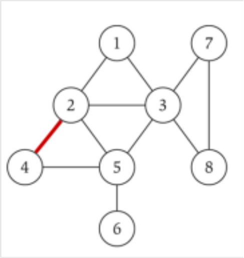
La matrice d'adjacence peut également être utilisée pour un graphe pondéré.

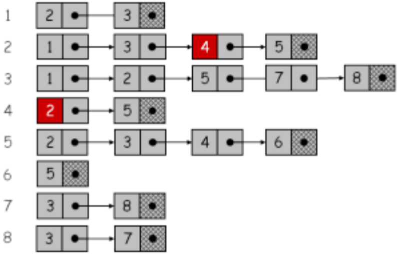
	A	B	C	D	E	F
A	0	0.5	∞	∞	∞	∞
B	0.5	0	7.7	∞	∞	∞
C	∞	0.5	0	0.1	∞	∞
D	∞	∞	0.1	0	0.4	0.5
E	∞	∞	∞	0.4	0	0.9
F	∞	∞	∞	∞	0.9	0



Listes d'adjacences

On peut aussi représenter un graphe simple en donnant pour chacun de ses sommets la liste des sommets auxquels il est adjacent. Ce sont les listes d'adjacences.

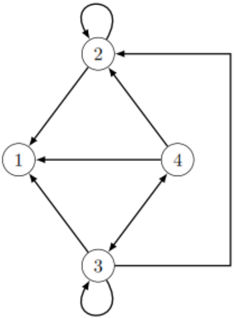




Application

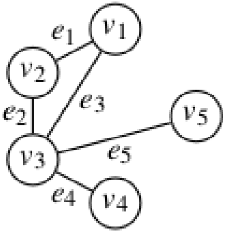
Soit le $G = (V, E)$ le graphe orienté suivant avec $V = \{1, 2, 3, 4\}$ et les arcs :

1. Donner la matrice d'adjacence de ce graphe.
2. Donner la liste d'adjacence associée à ce graphe.
3. Calculer les degrés entrant et sortant de chaque sommet.



Chaînes

Une **chaîne** dans G, est une suite de la forme $(v_0, e_1, v_1, e_2, \ldots, v_{k-1}, e_k, v_k)$ ayant pour éléments alternativement des sommets (v_i) et des arêtes (e_i), commençant et se terminant par un sommet, et telle que les extrémités de e_i soient v_{k-1} et v_i, $i = 1, \ldots, k$.
Si $v_0 = u$ et $v_k = v$, on dit que la chaîne relie u et v.
Une chaîne a une **longueur** k (c'est le nombre d'arêtes de la chaîne).
Une chaîne doit comporter **au moins une arête.**

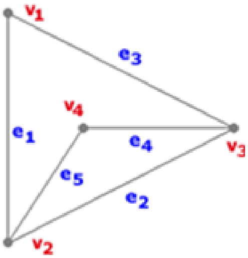


Chaînes

- On appelle **distance** entre deux sommets la longueur de la plus petite chaîne les reliant.
- On appelle **diamètre** d'un graphe la plus longue des distances entre deux sommets.
- Une chaîne **élémentaire** est une chaîne ne passant pas deux fois par un même sommet, c'est-à-dire dont tous les sommets sont distincts.
- Une chaîne **simple** est une chaîne ne passant pas deux fois par une même arête, c'est-à-dire dont toutes les arêtes sont distinctes.
- Une chaîne telle que $v_0 = v_k$ est appelée chaîne **fermée** ou **cycle**.

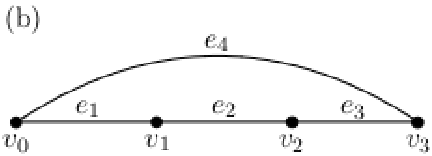
Application

- Quel est le diamètre du graphe ci-dessous ?
- Combien d'arêtes (et lesquelles) faut-il enlever pour que son diamètre soit de 3 ?



Cycles

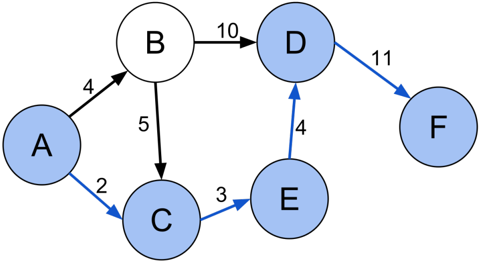
- Une chaîne fermée simple est appelée **cycle** si seul le sommet de départ apparaît deux fois dans la chaîne.
- Une boucle est un cycle de longueur 1.
- Un cycle élémentaire est un cycle dont les sommets sont **distincts**



Chemins

Définition

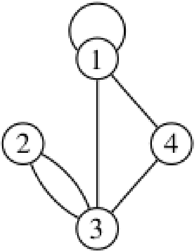
Un chemin est une chaîne telle que tous les arcs sont orientés dans le même sens.



Graphe simple

Définition

- Un graphe est **simple** s'il ne contient ni boucles ni arrêtes multiples.
- Si G est un graphe simple, le nombre d'arêtes incidentes à un sommet v de G est appelé **degré** de v dans G, noté $d_G(v)$ ou $d(u)$, avec $0 <= d(v) <= n - 1$.



Multigraphe

Graphe complet

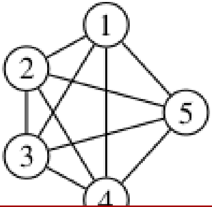
Définition

Le graphe G est dit un graphe **complet** à n sommets noté K_n si G est simple et toutes les paires de sommets sont directement reliées par une arête.

Remarque

Le nombre d'arêtes $m = n(n-1)/2$.

On appelle **clique** un sous-graphe complet de G.



Graphe complet K_5

$$V = \{1,2,3,4,5\}$$
$$E = \{\{1,2\},\{1,3\},\{1,4\},\{1,5\},\{2,3\},\{2,4\},\{2,5\},\{3,4\},\{3,5\},\{4,5\}\}$$

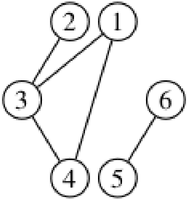
Graphe connexe

Définition

Un graphe est simplement connexe s'il existe une chaîne reliant une paire de sommets quelconques c.à.d à partir de n'importe quel sommet, il est possible de rejoindre tous les autres en suivant les arêtes.

Remarque

Un graphe non connexe se décompose en composantes connexes.



Graphe non connexe

$V = \{1, 2, 3, 4, 5, 6\}$

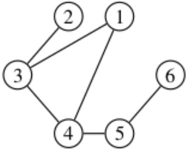
$E = \{\{1,3\}, \{1,4\}, \{2,3\}, \{3,4\}, \{5,6\}\}$

Graphe partiel

Un graphe $G' = (V, E')$ est un **graphe partiel** de $G(V, E)$, si obtient G' en enlevant une ou plusieurs arêtes au graphe G.

Sous-graphe

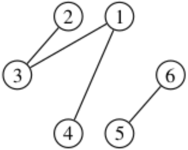
Un graphe $G' = (A, E(A))$ est un **sous-graphe** de G, si A est un sous-ensemble de sommets de V et $E(A)$ est formé de toutes les arêtes de G ayant leurs deux extrémités dans A.



Graphe G

$V = \{1,2,3,4,5,6\}$

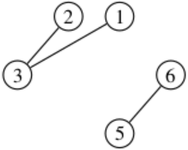
$E = \{\{1,3\},\{1,4\},\{2,3\},\{3,4\},\{4,5\},\{5,6\}\}$



Graphe partiel de G

$V = \{1,2,3,4,5,6\}$

$E = \{\{1,3\},\{1,4\},\{2,3\},\{5,6\}\}$



Sous-graphe de G

$V = \{1,2,3,5,6\}$

$E = \{\{1,3\},\{2,3\},\{5,6\}\}$

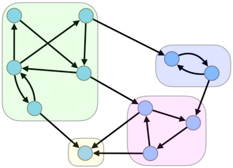
Composante connexe

Définition

La composante connexe d'un sommet s, notée $CC(s)$, est le sous-ensemble de sommets tels qu'il existe une chaîne entre deux sommets quelconques de $CC(s)$.

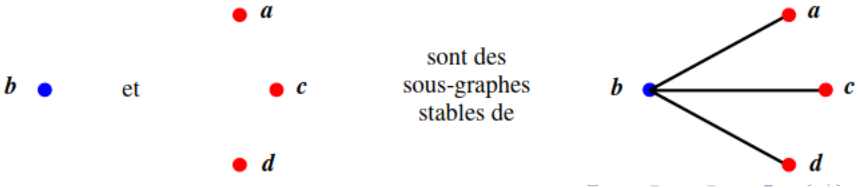
Définition

La composante fortement connexe d'un sommet d'un graphe orienté, notée $CFC(s)$, définit les sommets accessibles mutuellement par un chemin.



Stable

- Un sous-graphe est stable si ses sommets ne sont reliés par aucune arête.
- Un sous-ensemble S de V est un stable s'il ne comprend que des sommets non adjacents deux à deux.
- La taille d'un stable est égale au nombre de sommets qu'il contient. Un ensemble stable maximum est un ensemble stable de cardinalité maximum.
- Le nombre de stabilité d'un graphe noté $\alpha(G)$ est la cardinalité d'un stable maximum.



Graphes eulériens

Définition

- On dit qu'un graphe est eulérien s'il est possible de trouver un **cycle** passant une et une seule fois par toutes les arêtes.
- On dit qu'un graphe est **semi-eulérien** s'il est possible de trouver une **chaîne** passant une et une seule fois par toutes les arêtes.
- Plus simplement, on peut dire qu'un graphe est eulérien (ou semi-eulérien) s'il est possible de dessiner le graphe **sans lever le crayon (et sans passer deux fois sur le même trait!).**

Graphes eulériens

Theorem

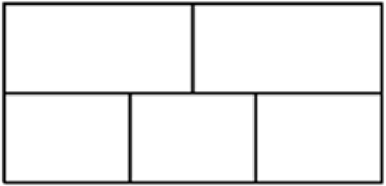
*Un graphe connexe G admet un cycle eulérien si et seulement si **tous ses sommets sont de degré pair**.*

Theorem

*Un graphe connexe G admet une chaîne eulérienne distincte d'un cycle si et seulement si **le nombre de sommets de** G **de degré impair est égal à 2**. Dans ce cas, si u et v sont les deux sommets de G de degré impair, alors le graphe G admet une chaîne eulérienne d'extrémités u et v.*

Application

Est-il possible de tracer une courbe, sans lever le crayon, qui coupe chacun des 16 segments de la figure suivante exactement une fois ?



Graphes hamiltoniens

Définition

- On dit qu'un graphe est hamiltonien s'il est possible de trouver un cycle passant une et une seule fois par tous les sommets.
- On dit qu'un graphe est semi-hamiltonien s'il est possible de trouver une chaîne passant une et une seule fois par tous les sommets.
- Un graphe possédant un sommet de degré 1 ne peut être hamiltonien.
- Si un sommet dans un graphe est de degré 2, alors les deux arêtes incidentes à ce sommet doivent faire partie du cycle hamiltonien.
- Les graphes complets K_n sont hamiltoniens.

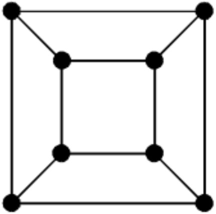
Graphes hamiltoniens

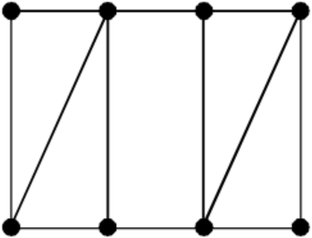
Théorème: Soit $G = (V, E)$ un graphe simple d'ordre $n >= 3$. Si pour toute paire (u, v) de sommets non adjacents, on a $d(u) + d(v) >= n$, alors G est hamiltonien.

Corollaire Soit $G = (V, E)$ un graphe simple d'ordre $n >= 3$. Si pour tout sommet u de G, on a $d(u) >= n/2$, alors G est hamiltonien.

Application

Montrer que ces deux graphes sont Hamiltoniens et non Eulériens

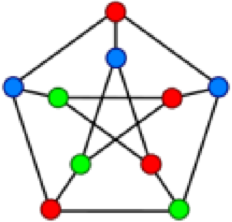




Chapitre2: Coloration

Coloration du graphe

La coloration des sommets d'un graphe consiste à affecter à tous les sommets de ce graphe une couleur de telle sorte que deux sommets adjacents ne portent pas la même couleur. Une coloration avec k couleurs est donc une partition de l'ensemble des sommets en k stables.

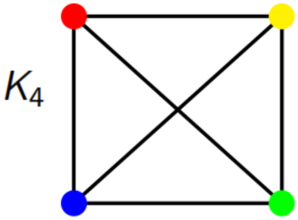


Coloration du graphe

Le nombre chromatique du graphe G, noté $\gamma(G)$, est le plus petit entier k pour lequel il existe une partition de V en k sous-ensembles stables.

Remarque

Le nombre chromatique de K_n est n.



Coloration du graphe

Theorem

$\chi(G) <= r + 1$, où r est le plus grand degré de ses sommets.
$\chi(G) <= n + 1 - \alpha(G)$, avec $\alpha(G)$ la cardinalité d'un stable S.

Theorem

Le nombre chromatique d'un graphe est supérieur ou égal à celui de chacun de ses sous-graphes.
$\chi(G) >= \omega(G)$: le nombre chromatique du graphe sera supérieur ou égal à l'ordre de sa plus grande clique (sous-graphe complet).

Exemple

Cinq étudiants A, B, C, D, E doivent passer les examens suivants :

A	Micro Economie, Anglais, Théorie des jeux
B	Probabilités, Analyse financière
C	Macro, Théorie des jeux
D	Anglais, Probabilités, Macro
E	Micro Economie, Analyse financière

Sachant que chaque étudiant ne peut se présenter qu'à une épreuve par jour, quel est le nombre minimal de jours nécessaire à l'organisation de toutes les épreuves ?

Algorithme de coloration de Welsh et Powell

Cet algorithme couramment utilisé permet d'obtenir une assez bonne coloration d'un graphe, c'est-à-dire une coloration n'utilisant pas un trop grand nombre de couleurs. Cependant il n'assure pas que le nombre de couleurs soit minimum (et donc égal au nombre chromatique du graphe).

Étape 1: Classer les sommets du graphe dans l'ordre décroissant de leur degré, et attribuer à chacun des sommets son numéro d'ordre dans la liste obtenue.

Étape 2: En parcourant la liste dans l'ordre, attribuer une couleur non encore utilisée au premier sommet non encore coloré, et attribuer cette même couleur à chaque sommet non encore coloré et non adjacent à un sommet de cette couleur.

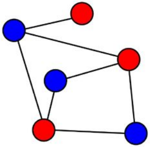
Étape 3: S'il reste des sommets non colorés dans le graphe, revenir à l'étape 2. Sinon, FIN.

Graphe biparti

On s'intéresse au cas des graphes non orientés dont le nombre chromatique est 2. Ils peuvent être définis ainsi:

Définition

Un graphe non orienté G est biparti s'il existe une partition de l'ensemble de ses sommets en deux classes disjointes telle que toute arête a une extrémité dans chaque classe.



Theorem

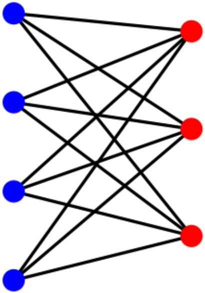
*Un graphe est **biparti** si et seulement s'il ne contient aucun cycle de **longueur impaire**.*

Graphe biparti complet

Définition

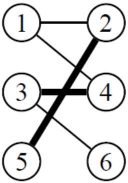
un graphe G est biparti complet si G est biparti et si tout sommet d'une partie est adjacent à tout sommet de l'autre partie. On note K le graphe biparti complet dont les deux parties ont respectivement p et r sommets.

$K_{4,3}$

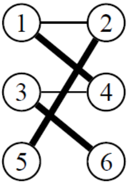


Couplage

- En regroupant les arêtes de même couleur nous obtenons une partition de l'ensemble des arêtes ; l'ensemble des arêtes ayant la même couleur forme un couplage.
- On appelle **couplage** d'un graphe G, un sous-ensemble arrêtes de G deux à deux non adjacentes.
- **Couplage parfait (Perfect matching)** Dans un graphe à 2n sommets, un couplage avec n arêtes est dit parfait. Chaque sommet du graphe est saturé par un couplage parfait.



Un couplage



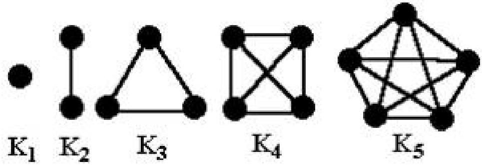
Un couplage maximum et parfait

Graphes planaires

Définition

Un graphe est **planaire** s'il peut être dessiné sur un plan sans que les arêtes ne s'intersectent (en dehors des sommets).

- Par exemple, considérons les cinq graphes suivants, quels graphes son-ils planaires? :

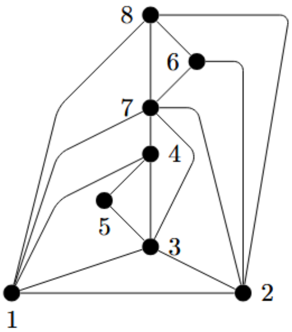


Le nombre chromatique $\chi(G)$ d'un graphe planaire est au plus 4.

Graphes planaires

Définition

Donner une coloration minimale du graphe suivant:



Chapitre3:Arbre/Arborescence

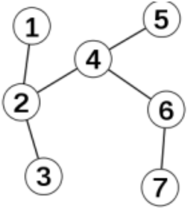
Arbre

Définition

On appelle arbre tout graphe connexe sans cycle.
Un graphe sans cycle mais non connexe est appelé une forêt.

On distingue deux types de sommets dans un arbre :

- Les feuilles dont le degré est 1;
- Les sommets internes dont le degré est supérieur à 1.



Arbre

Theorem

Les affirmations suivantes sont équivalentes pour tout graphe G à n sommets:

- *G est un arbre*
- *G est sans cycle et connexe*
- *G est sans cycle et comporte $n - 1$ arêtes*
- *G est connexe et comporte $n - 1$ arêtes chaque paire u, v de sommets distincts est reliée par une seule chaîne simple (et le graphe est sans boucle).*

Arborescence

Définition

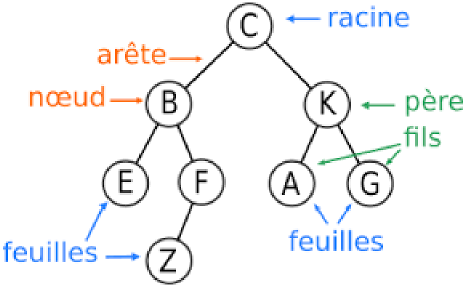
Une arborescence est un arbre orientée G dont tous les sommets sont de degrée entrant égalé à 1 à l'exception d'un seul, appelé racine, pour lequel le degré entrant est nul.

- Dans une arborescence, tout sommet s_j accessible depuis s_i est désigné descendant de s_i tandis que s_i est désigné ancêtre de s_j .
- Soit un arc a reliant le sommet s_i au sommet s_j , s_i est désigné père de s_j et s_j le fils de s_i.
- Si deux sommets ont le même père, on dit qu'ils sont frères.

Arborescence

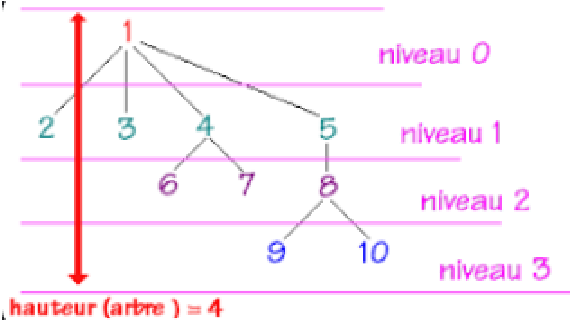
Dans une arborescence définit, il existe trois types de sommets:

- la racine (sans père),
- les feuilles (sans fils),
- les nœuds (un père et au moins un fils).



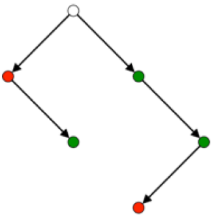
Arborescence

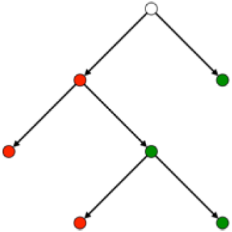
- Le nombre de fils d'un sommet est appelé le degré de ce sommet
- La longueur du chemin entre la racine de l'arborescent et un sommet définit la profondeur de ce sommet.
- La plus grande profondeur détermine la hauteur de l'arborescent.



Arborescence binaire

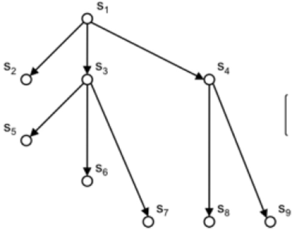
- Une arborescence binaire est une arborescence où chaque sommet a au plus deux fils:
  - un fils gauche
  - un fils droit
- Dans une arborescence binaire, la position du fils est importante: gauche ou droite.
- Une arborescence binaire est complète, si chaque sommet qui a des fils, a un fils gauche et un fils droit.





Arborescence binaire

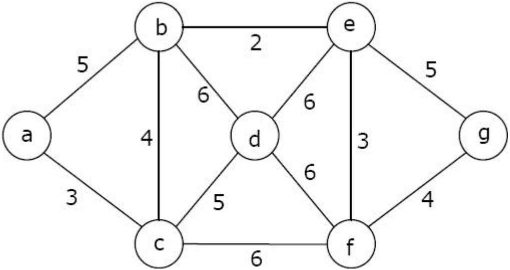
Une arborescence peut être représentée par un tableau. Le ieme élément du tableau définit le sommet père du sommet *i* la racine de l'arborescence est représentée par le symbole ˆ.



S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
^	S_1	S_1	S_1	S_3	S_3	S_3	S_4	S_4

Motivation

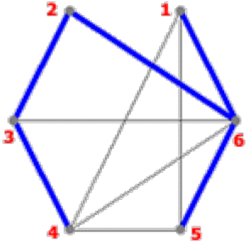
Un réseau comporte des machines a, b, c, d, e, f, et g qui doivent pouvoir communiquer entre elles. Les liaisons envisagées sont Représentées par le graphe suivant (les arêtes sont étiquetées par la distance entre les machines) :

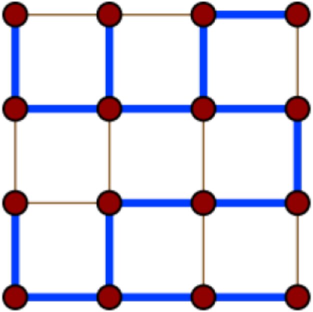


Question : Comment câbler le réseau à moindre coût?

Définition

Un arbre couvrant (Spanning Tree) du graphe G est un graphe partiel qui est aussi un arbre.





Arbre couvrant

Algorithmes de détermination d'un arbre couvrant

Deux idées duales :

$A := (X, \emptyset)$ (le graphe vide)
tant que $nbAretes(A) < V - 1$ **faire**
- Choisir une arête de G qui ne crée pas de cycle.
- Ajouter cette arête à A.

$A := (X, R)$ (le graphe G)
tant que $nbAretes(A) > V - 1$ **faire**
- Choisir une arête de A qui n'est pas indispensable à la connexité.
- Retirer cette arête à A.

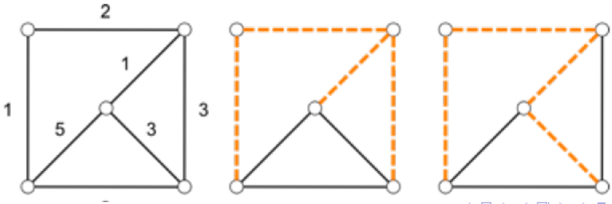
Par construction le graphe A obtenu est un arbre couvrant (pour peu que G soit connexe).

Critère de choix

- ne pas créer de cycle : assez facile
- vérifier la connexité : moins facile (algorithme **Reverse-Delete**)

Arbres couvrants minimaux

Le problème de l'arbre couvrant de poids minimum (Minimum Spanning Tree) consiste à trouver un arbre couvrant du graphe tel que son poids soit minimum. Ce problème se pose, par exemple, lorsqu'on désire relier n villes par un réseau routier de coût minimum. Les sommets du graphe Représentent les villes, les arêtes, les tronçons qu'il est possible de construire et les poids des arêtes correspondent aux coûts de construction du tronçon correspondant. L'exemple suivant montre que la solution n'est pas unique :



Arbres couvrants minimaux

Choisir une arête

Toute arête qui ne crée pas de cycle convient.
Pour que l'arbre soit minimal, il faut aussi tenir compte des poids.

Là encore, deux politiques :

Connexité d'abord : Algorithme de Prim

On choisit l'arête de poids minimal **parmi celles incidentes à** A.
En cours d'algorithme A est un **arbre**.

Il suffit qu'une extrémité de l'arête choisie soit hors de A.

Minimalité d'abord : Algorithme de Kruskal

On choisit l'arête de poids minimal **dans tout le graphe**.
En cours d'algorithme A est une **forêt**.

Il faut mémoriser si deux sommets sont dans des **composantes connexes** distinctes : utilisation de *Union-Find*.

Algorithmes gloutons

Algorithme Kruskal

Principe

- L'algorithme construit un arbre couvrant minimum en sélectionnant des arêtes par poids croissant.
- L'algorithme considère toutes les arêtes du graphe par poids croissant (en pratique, on trie d'abord les arêtes du graphe par poids croissant) et pour chacune d'elles, il la sélectionne si elle ne crée pas un cycle.

Remarque: On remarque que lors du déroulement de l'algorithme, les arêtes sélectionnées ne forment pas nécessairement un graphe connexe. Mais à la fin, les arêtes sélectionnées forment un graphe connexe.

Chapitre4:Exploration d'un graphe

Motivation: théorie des jeux

Deux joueurs, A et B, choisissent tour à tour un chiffre, soit 1 soit 2. Le joueur A commence. Une fois que quatre chiffres ont été choisis, la partie se termine, et le plus grand diviseur premier du nombre formé par ces quatre chiffres, écrits dans l'ordre de sélection de gauche à droite, détermine le gain de B (et la perte de A). Expliquez ce qu'est une stratégie optimale pour un joueur et déterminez une telle stratégie pour B, en précisant le gain minimal garanti par celle-ci.

Recherche d'un parcours

- **Graphe non orienté:** Identifier les sommets accessibles depuis un sommet donné d'un graphe. En d'autres termes, il s'agit d'identifier toutes les chaînes contenant le sommet choisi.
- **Graphe orienté:** Identifier tous les chemins qui ont pour origine le sommet choisi.

Cette recherche de parcours est un problème fréquent de la théorie de parcours, elle peut s'effectuer de deux manières:

- parcours en largeur BFS (Breadth First Search)
- parcours en profondeur DFS (Depth First Search)

Les résultats de ces parcours sont des arborescences qui identifient les sommets accessibles depuis un sommet donné.

Méthode de recherche d'un parcours en largeur

Le résultat de ce parcours est formalisé d'une part par une arborescence, désignée arborescence en largeur, contenant tous les sommets accessibles depuis un sommet donné et, d'autre part, par un tableau mémorisant les distances en arc.

- Le tableau, noté A_L, représente l'arborescence en largeur
- Le tableau, noté Δ_L, mémorise les distances en arc.

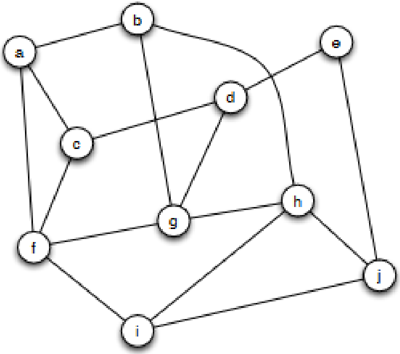
Méthode de recherche d'un parcours en largeur

L'algorithme

1. Initialiser $SO = \emptyset$, $A_L[s] = --$ et $\triangle_L[s] = \infty$ pour tous les sommets du graphe
2. Choisir un sommet s_i du graphe
    1. $A_L[s_i] = \hat{}$
    2. $\triangle_L[s] = 0$
    3. Ajouter s_i à SO
3. Rechercher les sommets adjacents (vers l'extérieur dans un graphe orienté) à SO Pour tous les sommets s_k trouvés:
    1. $A_L[s_k] = predecesseur[s_k]$
    2. $\triangle_L[s_k] = \triangle_L[predecesseur[s_k]] + 1$
    3. Ajouter les sommets s_k à SO
4. Recommencer la recherche jusqu'à ce qu'il n'existe plus de sommets adjacents (vers l'extérieur dans un graphe orienté) à SO.

Application

Construisez l'arbre de parcours en largeur (BFS) en considérant le sommet ""***a*** comme racine.



Méthode de recherche d'un parcours en profondeur

- **Le parcours en profondeur** est un parcours qui cherche à descendre plus profondément dans le graphe chaque fois que c'est possible.
- Ce parcours construit une forêt en profondeur composée de plusieurs arborescences en profondeur.

DFS

Algorithme

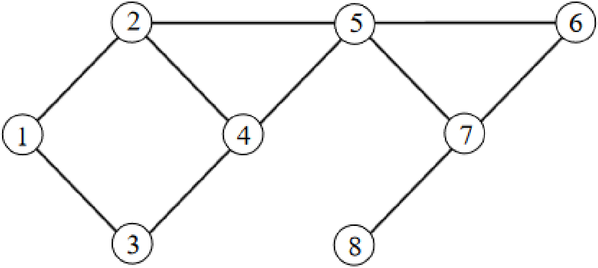
Les étapes de la construction de ce parcours sont les suivantes:

1. Etape initiale : Tous les sommets sont coloriés en Couleur1.
2. Etape 1 : Un sommet colorié en Couleur1 est choisi arbitrairement et il est colorié en Couleur2 et devient le sommet courant. Ce sommet est racine dune arborescence en profondeur.
3. Etape 2 : Si le sommet courant a un sommet successeur (ou voisin) colorié en Couleur1, on le colorie en Couleur2 et on ajoute ce sommet dans l'arborescence en profondeur et il devient le sommet courant.
4. Etape 3 : S'il n'existe aucun sommet successeur Couleur1, le sommet est marqué en Couleur3 et on remonte au sommet père dans l'arborescence en profondeur qui devient le sommet courant.

Ce parcours reprend à partir de l'étape 2 jusqu'à ce que tous les sommets accessibles à partir du sommet défini à l'étape 1 aient été découverts. A la fin du parcours, s'il reste des sommets coloriés en Couleur1 on recommence à l'étape 1.

Application

1. Donner un parcours DFS de G et une arborescence associée.
2. Est-ce que les listes $L1 = (5, 6, 2, 7, 8, 1, 3, 4)$ et $L2 = (5, 6, 3, 2, 1, 4, 8, 7)$ sont des parcours ?



Remarques: BFS/DFS

- **BFS**: Permet de décider si le graphe G est **connexe**. G est connexe si et seulement si $\Delta_L[s, v] < \infty; \forall v \in V$.
- **BFS**: Donne un seul arbre, mais il peut y en avoir plusieurs. cela dépend de l'ordre dans lequel les sommets sont considérés.
- **DFS**: Permet de vérifier rapidement la connexité du graphe. Si le graphe est connexe, un parcours DFS à partir d'un sommet couvrira tous les sommets du graphe.
- **DFS**: est utilisé pour détecter les composantes fortement connexes dans un graphe orienté.

Définition

La recherche des plus courts chemins dans un graphe est un problème classique de la théorie des graphes. Il est possible d'identifier deux classes de problèmes :

- la recherche des plus courts chemins depuis une origine unique,
- la recherche des plus courts chemins pour tout couple de sommets.

Définition

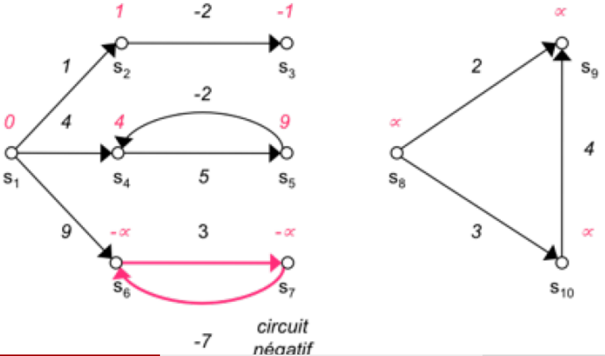
Ce type de problème traite un graphe orienté doté d'une fonction de Pondération à valeurs réelles.
Le poids d'un chemin $C_{ij} = (a_m; a_n; :::; a_k)$, noté $P(C_{ij})$, est la somme des poids des arcs qui le constituent.
Le poids du plus court chemin entre s_i et s_j est défini par

$$f(x) = \begin{cases} \Delta_P(si; sj) = min(P(C_{ij})), & \text{où } C_{ij} \text{ est un chemin entre si et sj} \\ \infty, & \text{s'il n'existe pas de chemin entre si} \end{cases} \quad (1)$$

Un plus court chemin d'un sommet s_i et s_j est alors défini comme un chemin C_{ij} de poids $\Delta_P(si; sj)$.

Existence d'un plus court chemin

Si le graphe ne contient aucun circuit de poids négatif accessible à partir d'un sommet s_i, alors pour tout sommet s_j du graphe le problème du plus court chemin$\Delta_P(si; sj)$ reste défini même si sa valeur est négative.



Existence d'un plus court chemin

Theorem

Les sous-chemins des plus courts chemins sont des plus courts chemins: Soit C_{ij} *un plus court chemin d'un sommet* s_i *à un sommet* s_j *et soit* C_{qr} *un sous-chemin de* C_{ij} *allant du sommet* s_q *au sommet* s_r*, alors* C_{qr} *est un plus court chemin de* s_q *et* s_r*.*

Plus court chemin à origine unique

Dans ce type de problème, il s'agit d'une part d'identifier les sommets accessibles depuis un sommet d'origine s_0 et d'autre part, pour chaque sommet, de définir le plus chemin pour atteindre ce sommet depuis s_0.
Le résultat de ce problème est représenté par une arborescence, désignée **arborescence de plus court chemin**.
Il existe deux méthodes pour construire une arborescence de plus court chemin depuis un sommet :

- la méthode de Dijkstra,
- la méthode de Bellman-Ford.

Méthode de Dijkstra

Cette méthode s'applique uniquement si tous les arcs ont un poids positif ou nul. Les étapes de cette méthode sont les suivantes:

- **Étape 1** Poser $S_1 = \emptyset$ et $S_2 =$V.
- **Étape 2** Choisir un sommet s_i et initialiser les estimations: $E[s_i] = 0$ et $E[s_j] = \infty$ pour tous les sommets s_j appartenant à $V - s_i$.
- **Étape 3** Rechercher dans S_2 le sommet dont l'estimation est minimale.
- **Étape 4** Ajouter le sommet retenu à l'étape 3 dans S_1 et le retirer de S_2.
- **Étape 5** Relâcher tous les arcs traversant vers l'extérieur la coupure $(S_1; S_2)$.

Si S_2 est vide le traitement est terminé sinon il faut recommencer à l'étape 3.

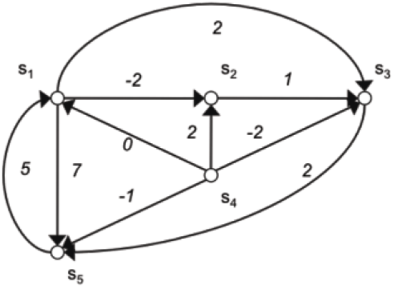
Méthode de Bellman-Ford

Cette méthode s'applique uniquement si tous les arcs ont un poids positif ou nul ou poids négatifs.

- **Étape 1** Choisir un sommet s_i
- **Étape 2** Initialiser les estimations : $E[s_i] = 0$ et $E[s_j] = \infty$ pour tous les sommets s_j appartenant à $S - s_i$.
- **Étape 3** Relâcher tous les arcs du graphe $|S| - 1$ fois.

Plus court chemin pour tout couple de sommets

Dans le cadre de cette étude, nous considérons qu'il n'existe pas de circuit de poids négatif au sein des graphes.
Dans le cas d'un graphe pondéré, la matrice d'adjacence MA permet de représenter le poids des arcs entre deux sommets, ou ∞ s'il n'existe plus d'un arc .



$$M = \begin{pmatrix} \infty & -2 & 2 & \infty & 7 \\ \infty & \infty & 1 & \infty & \infty \\ \infty & \infty & \infty & \infty & 2 \\ 0 & 2 & -2 & \infty & -1 \\ 5 & \infty & \infty & \infty & \infty \end{pmatrix}$$

Méthode algébrique

La méthode algébrique peut être schématisée comme suit:

- **Étape 1** Déterminer $M = MA$ avec 0 sur la diagonale (élimination des boucles si elles existent)
- **Étape 2** Initialisation du compteur :$i \leftarrow 2$
- **Étape 3** Calculer de $M^i = M^{i-1} \times M$

Si $M^i \neq M^{i-1}$ et $i <\mid S \mid -1$, on incrémente i et on recommence à l'étape 3.

Méthode algébrique

Exemple:

Calcul de M^2:

$$\begin{pmatrix} 0 & -2 & 2 & \propto & 7 \\ \propto & 0 & 1 & \propto & \propto \\ \propto & \propto & 0 & \propto & 2 \\ 0 & 2 & -2 & 0 & -1 \\ 5 & \propto & \propto & \propto & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & -2 & 2 & \propto & 7 \\ \propto & 0 & 1 & \propto & \propto \\ \propto & \propto & 0 & \propto & 2 \\ 0 & 2 & -2 & 0 & -1 \\ 5 & \propto & \propto & \propto & 0 \end{pmatrix} \quad \begin{pmatrix} \mathbf{0} & \mathbf{-2} & \mathbf{-1} & \propto & \mathbf{4} \\ \propto & \mathbf{0} & \mathbf{1} & \propto & \mathbf{3} \\ \propto & \propto & \mathbf{0} & \propto & \mathbf{2} \\ \mathbf{0} & \mathbf{-2} & \mathbf{-2} & \mathbf{0} & \mathbf{-1} \\ \mathbf{5} & \mathbf{3} & \mathbf{7} & \propto & \mathbf{0} \end{pmatrix}$$

Méthode algébrique

Exemple

Calcul de M^3:

$$\begin{pmatrix} 0 & -2 & 2 & \propto & 7 \\ \propto & 0 & 1 & \propto & \propto \\ \propto & \propto & 0 & \propto & 2 \\ 0 & -2 & -2 & 0 & -1 \\ 5 & \propto & \propto & \propto & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & -2 & -1 & \propto & 4 \\ \propto & 0 & 1 & \propto & 3 \\ \propto & \propto & 0 & \propto & 2 \\ 0 & -2 & -2 & 0 & -1 \\ 5 & 3 & 7 & \propto & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & -2 & -1 & \propto & 1 \\ 8 & 0 & 1 & \propto & 3 \\ 7 & 5 & 0 & \propto & 2 \\ 0 & -2 & -2 & 0 & -1 \\ 5 & 3 & 4 & \propto & 0 \end{pmatrix}$$

Méthode algébrique

Calcul de M^4:

$$\begin{pmatrix} 0 & -2 & 2 & \propto & 7 \\ \propto & 0 & 1 & \propto & \propto \\ \propto & \propto & 0 & \propto & 2 \\ 0 & 2 & -2 & 0 & -1 \\ 5 & \propto & \propto & \propto & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & -2 & -1 & \propto & 1 \\ 8 & 0 & 1 & \propto & 3 \\ 7 & 5 & 0 & \propto & 2 \\ 0 & -2 & -2 & 0 & -1 \\ 5 & 3 & 4 & \propto & 0 \end{pmatrix} \quad \begin{pmatrix} \mathbf{0} & \mathbf{-2} & \mathbf{-1} & \propto & \mathbf{1} \\ \mathbf{8} & \mathbf{0} & \mathbf{1} & \propto & \mathbf{3} \\ \mathbf{7} & \mathbf{5} & \mathbf{0} & \propto & \mathbf{2} \\ \mathbf{0} & \mathbf{-2} & \mathbf{-2} & \mathbf{0} & \mathbf{-1} \\ \mathbf{5} & \mathbf{3} & \mathbf{4} & \propto & \mathbf{0} \end{pmatrix}$$

$M^4 = M^3$ signifie que la matrice s'est **stabilisée**. Tous les chemins les plus courts ont été trouvés en 3 étapes ou moins. Toute multiplication ultérieure (M^5, M^6, etc.) donnera la même matrice.