I) Notions générales

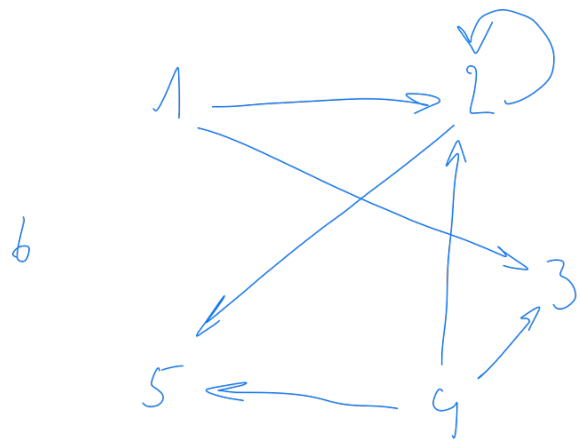
1- Définition

Définition mathématique:
Un graphe $G = (X, U)$ est un ensemble fin X de points, appelés noeuds ou sommets, associés à un ensemble fini U de lignes appelées arc ($U \subset X \times X$) (U est formé de couple de X).

Passage à la représentation géométrique :

- à chaque sommet $x \in X$, on associe un point du plan
- à chaque arc (x_i, x_j) où x_i et $x_j \in X$, et associe une ligne fléchée qui va de x_i à x_j

Exemple 1:
$X = \{1,2,3,4,5,6\}$
$U = \{(1,2), (1,3), (2,2), (2,5), (4,2), (4,3), (4,5)\}$



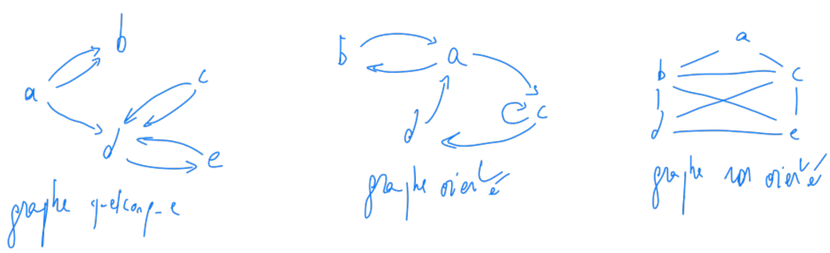
2- Graphe orienté

Un graphe où il y a au plus un arc joignant un sommet de départ et un sommet d'arrivé donnés. Si a et b sont deux noeuds, l'arc ab (s'il existe) est différent de l'arc ba (s'il existe)

3- Graphe non orienté

Un graphe où la distinction entre départ et arrivée disparaît: l'arc ab et l'arc ba ne peuvent exister qu'en même temps, et sont alors confondus. Il y a au moins plus une <u>arête</u> (équivalent d'arc) joignant deux noeuds donnés.

Exemple:



4- Graphe symétrique (non orienté)

Un graphe est symétrique si $\forall$ arc $u \in U, \exists$ un arc $u' \in U$ tel que:

$I(u) = T(u')$ et $\qquad I$: Initial

$T(u) = I(u')$ $\qquad T$: Terminal

Exemple 3:

5- Graphe simple

Un graphe tel que, entre deux sommets il ne peut y avoir qu'un seul arc dans le même sens.

Exemple 4: le graphe orienté ci-dessus est un graphe simple alors que le graphe quelconque ne l'est pas.

6- Graphe généralisé (ou quelconque)

C'est un graphe qui n'est pas simple. On rencontre dans ces graphes des situations telles que:

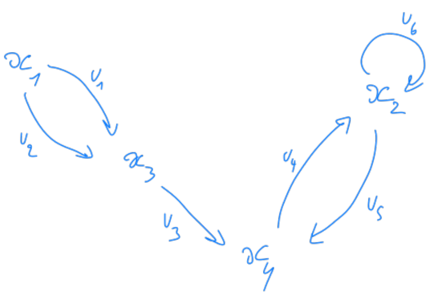
7- Multiplicité d'un arc

Soit $G = (X, U)$ un graphe. On définit la multiplicité de l'arc (x, y), où x et $y \in X$, de la façon suivante:

$$m_G = |\{u \in U / I(u) = x \text{ et } T(u) = y\}|$$

Exemple 5:

$$m_G(x_1, x_3) = 2$$
$$m_G(x_4, x_2) = 1$$
$$m_G(x_1, x_2) = 0$$



8- Extrémité

Soit $G = (X, U)$ un graphe simple, si $u = (x, y) \in U$ (u arc de G) alors $x \in X$ et $y \in X$. On dit que:

x est l'extrémité initiale (origine) de u, $x = I(u)$

y est l'extrémité terminale (terminaison) de u, $y = T(u)$

si $x = y$ on dit que u est une boucle

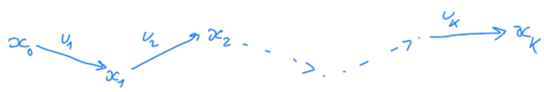
9- Chemin (chaîne pour un graphe symétrique)

Soient x_0 et $x_k \in X$. Un chemin de x_0 à x_k est une suite d'arc (si elle existe): $u_1, u_2, \ldots, u_k \in U$ telle que:

$$I(u_1) = x_0$$
$$I(u_i) = x_{i+1},\ 1 < i < k$$
$$T(u_k) = x_k$$

x_0 est l'extrémité initiale du chemin

x_k est l'extrémité terminale du chemin



Remarque: le chemin se note aussi $C = (x_0, u_1, x_2, u_2, \ldots, u_k, x_k)$

Exemple: dans le graphe de l'exemple 1, il existe des chemin allant de 1 à 5 mais il n'existe pas de chemin allant de 4 à 1

10- Chemin simple

Un chemin simple ne comporte pas deux fois le même arc.

11- Chemin élémentaire

Un chemin est élémentaire s'il ne passe pas deux fois par le même sommet.

Propriété:
Pour tout chemin f, il existe un chemin élémentaire f' ayant même origine et même terminaison.
Preuve:
On enlève les circuits contenus dans f.

Propriété:
Un chemin élémentaire est de longueur $\leq$ card$(X) - 1$
Exercice: Démontrer cette propriété

12- Longueur d'un chemin

C'est le nombre d'arc.

13- Circuit (cycle pour un graphe symétrique)

C'est un chemin tel que $x = x_k$ $\qquad (I(u_1) = T(u_k))$

14- Circuit simple

Un circuit est simple s'il ne comporte pas deux fois le même arc.

15- Circuit élémentaire

Un circuit est élémentaire s'il ne passe pas deux fois par le même sommet.

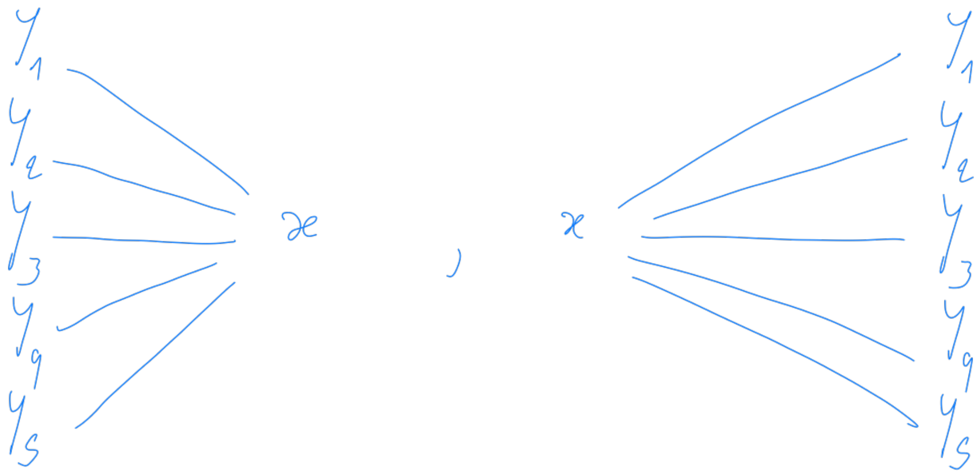
16- Prédécesseur

Soit $G = (X, U)$ un graphe, et soient x et $y \in X$. y est <u>prédécesseur</u> de x dans G, s'il existe $u \in U$ tel que $x = I(u)$ et $y = T(u)$

17- Successeur

Soit $G = (X, U)$ un graphe, et soient x et $y \in X$. y est successeur de x dans G, s'il existe $u \in U$ tel que $x = I(u)$ et $y = T(u)$

Exemple 8:



y_i avec $(1 \leq i \leq 5)$ est prédécesseur de x dans le premier cas
y_i avec $(1 \leq i \leq 5)$ est successeur de x dans le premier cas

18- Ascendant

Soit $G = (X, U)$ un graphe, et soient x et $y \in X$. y est ascendant de x dans G, s'il existe un chemin d'origine y et d'extrémité x.

19- Descendant

Soit $G = (X, U)$ un graphe, et soient x et $y \in X$. y est descendant de x dans G, s'il existe un chemin d'origine x et d'extrémité y.

20- Adjacence

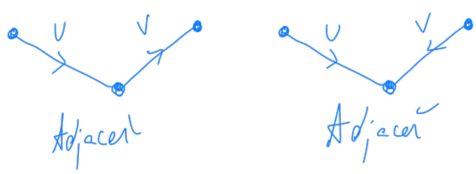
Soit $G = (X, U)$ un graphe, et $u = (x, y)$ un arc de U. On dit que:

- l'arc u est adjacent aux sommets x et y
- les sommets x et y sont adjacents à l'arc u
- les sommets x et y sont adjacents entre eux

21- Adjacent entre arcs

Deux arcs sont adjacents s'il on une extrémité commune.

Exemple 9:



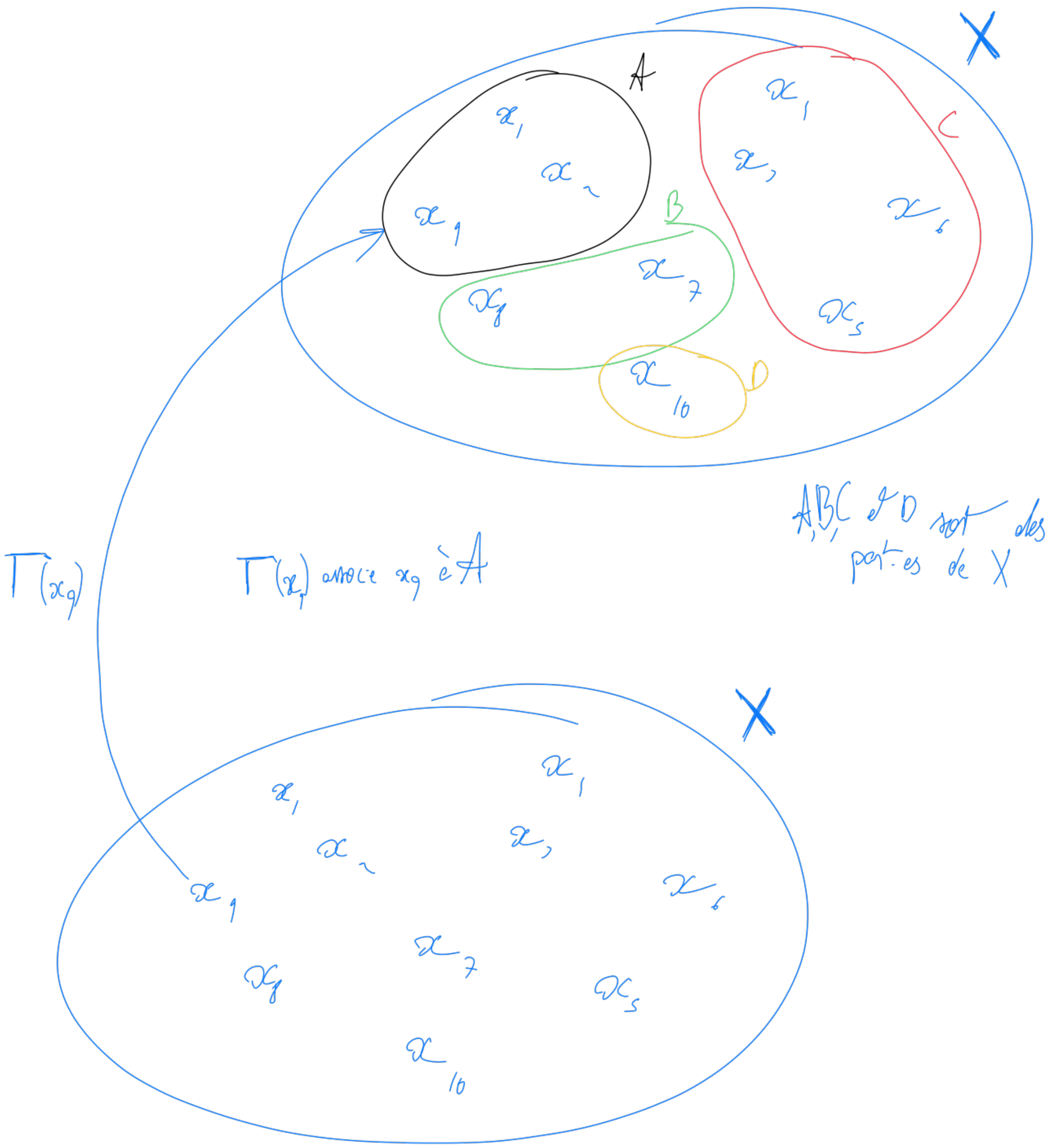
22- Degré d'un sommet

C'est le nombre d'arcs dont le sommets est l'une des extrémités.
Soit x un sommet du graphe $G = (X, U)$.
$Degré(x) = |\{u \in U / I(u) = x$ ou $T(u) = x\}|$

23- Fonction successeur

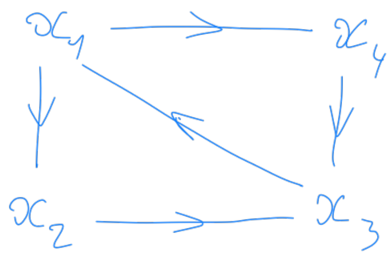
Soit un graphe $G = (X, U)$. On défini sur G une fonction successeur :

$\Gamma : X \longrightarrow \mathscr{P}(X)$ ($\mathscr{P}(X)$: l'ensemble des parties de X)



$\Gamma(x) = \{y \in X / \exists u \in U / u = (x, y)\}$

Exemple 10:



$\Gamma(x_1) = \{x_2, x_4\}$
$\Gamma(x_2) = \{x_3\}$
$\Gamma(x_3) = \{x_1\}$
$\Gamma(x_4) = \{x_3\}$

24- Fonction prédécesseur

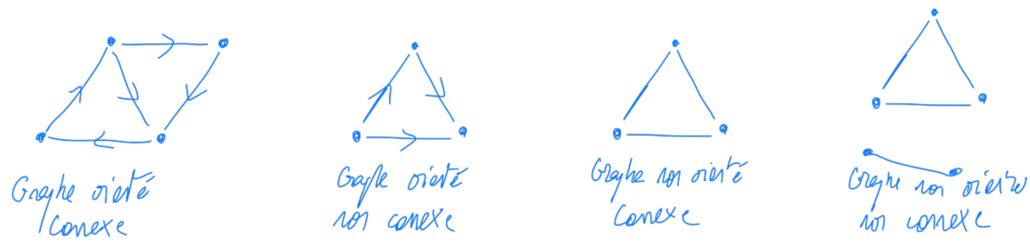
Soit un graphe $G = (X, U)$. On défini sur G une fonction prédécesseur:

$$\Gamma^{-1} : X \longrightarrow \mathscr{P}(X)$$
$$\Gamma^{-1}(x) = \{y \in X / x \in \Gamma(y)\}$$

25- Connexité

Un graphe est connexe si pour tout couple (x, y) de sommet distinct à X, il existe au moins un chemin de longueur finie ayant pour origine x et pour terminaison y.

Exemple 11:



26- Forte connexité

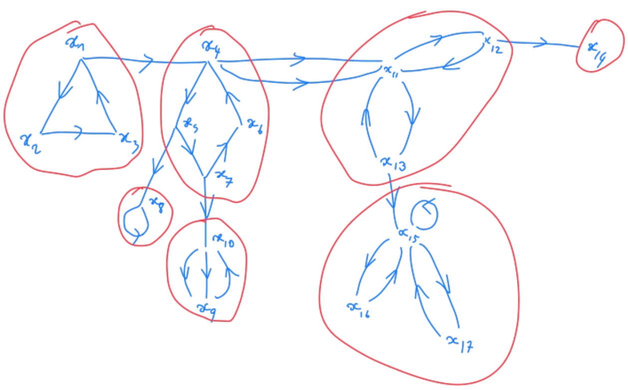
Soit $G = (X, U)$ un graphe.
La connexité forte est une relation d'équivalence $\hat{C}$ sur les sommets du graphe G définie par:

$$x\hat{C}y \Leftrightarrow \begin{cases} x = y \\ \text{ou bien} \\ \exists \text{ un chemin allant de } x \text{ à } y \text{ et un chemin allant de } y \text{ à } x \end{cases}$$

Cette relation est réflexive, symétrique et transitive.

Une relation d'équivalence sur un ensemble, partage cet ensemble en classe d'équivalence. Ici, les classes d'équivalence sont les composantes fortement connexes, on dit que le graphe est fortement connexe.

Exemple 12:



O : composante fortement connexe

Le graphe réduit $G = (X, U)$ est le graphe simple $= (\hat{X}, \hat{U})$ défini par:

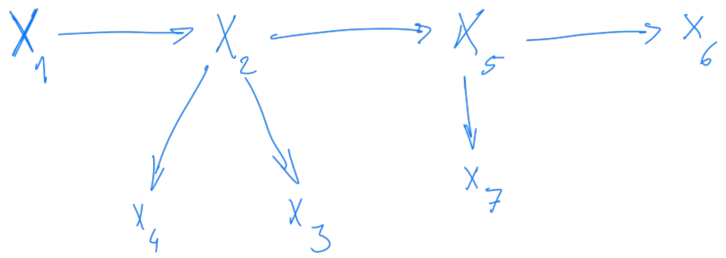
$\hat{X}$: Ensemble des composantes fortement connexes de G

$\hat{X}$: $\{X_1, X_2, \ldots, X_n\}$ ensemble des classes d'équivalence de X pour $\hat{C}$

$(X_i, x_j) \in \hat{U}$ s'il existe dans G un chemin allant d'un représentant de X_i à un représentant de X_j

$(X_i, x_j) \in \hat{U}$ si et seulement si il existe $(x, y) \in U$ où $\begin{cases} x \in X \\ y \in X \end{cases}$

Exemple: Pour le graphe de l'exemple précédent, on obtient le graphe réduit suivant:



II) Représentation des graphes

Soit $G = (X, U)$ un graphe, on peut le représenter de plusieurs façons différentes.

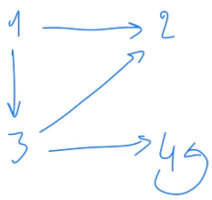
1- Matrice d'adjacence

C'est une matrice d'ordre $||X|| \times ||X||$ n'ayant que des 0 ou des 1.
Soit A la matrice d'adjacence d'un graphe,

$$A(i, j) = \begin{cases} 1 \text{ si et seulement si } \exists \text{ un arc du noeud } i \text{ au noeud } j \\ 0 \text{ sinon} \end{cases}$$

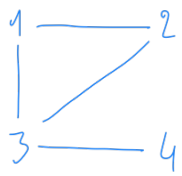
Exemple 14:

G1



$$A1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

G2



$$A2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Remarque: Pour un graphe non orienté la matrice d'adjacence est fortement symétrique

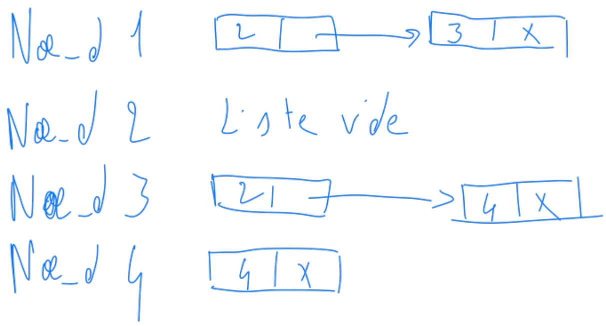
Avantage d'une telle représentation:
La détermination de la présence d'un arc est effectuée rapidement, en un temps qui ne dépend pas de $\|U\|$, ni de $\|X\|$

Inconvénient d'une telle représentation:
Le plus inconvénient est qu'elle nécessite $\|X\|^2$ mémoires

2- Liste d'adjacence

A un nœud u, on associe la liste de tous les nœuds v adjacents à u

Exemple 15: Pour le graphe G1, on avait les listes suivantes:

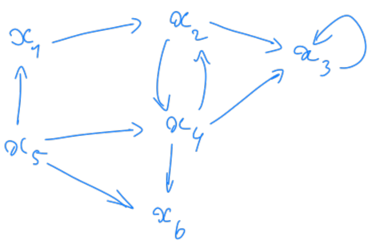


(Et on utilise une case supplémentaire pour lier les noeuds entre eux, en gros ça forme un tableau à deux dimensions)

3- Tableau de successeur

Le tableau indique les successeurs d'un sommet donné
La liste des successeurs d'un sommet se termine par 0
Quand il y a deux 0 qui se suivent alors le sommet n'a pas de successeurs
Pour trouver les successeurs d'un sommet, on recherche le $(i-1)^{\text{ème}}$ zéro

Exemple 16:



TabSuc

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
x_1	0	x_3	x_4	0	x_3	0	x_2	x_6	x_3	0	x_1	x_4	x_6	0	0

Pour avoir les successeurs du sommets x_4, on cherche le troisième 0 dans le tableau TabSuc

Le nombre d'éléments (valide) tableau est de N + M
N: nombre d'arcs
M: nombre de sommets

Remarque: On utilise ce type de représentation quand le graphe n'a pas beaucoup d'arcs

4- Tableau de premiers successeurs et tableau de successeurs

Utilisation d'un tableau (Premsuc) donnant l'indice du premier successeur (désigné comme tel) d'un sommet dans TabSuc
Utilisation d'un tableau (TabSuc) donnant la liste des successeurs d'un sommet, chaque liste est séparée par un 0

Exemple 17: Pour le graphe précédent, on a les deux tableaux suivants:

(Y'a peut-être un truc ici)

III) Algorithme sur les graphes

1- Recherche d'un arc
Soit un graphe $G = (X, U)$ et soient x et $y \in X$
Question: a-t-on $u = (x, y) \in U$?

2- Suppression d'un arc
Soit un graphe $G = (X, U)$ et soit $u = (x, y) \in U$
On veut supprimer l'arc qui relie le sommet x au sommet y

3- Adjonction d'un arc
Soit un graphe $G = (X, U)$
On veut rajouter au graphe l'arc $u = (x, y)$

Exercice: écrire les algorithmes 1- 2- 3- en utilisant les structures de données présentée ci-dessus

4- Parcours en largeur
Données: Un graphe $G = (X, U)$ orienté et un sommet x
Résultat: On veut avoir la liste de tous les sommets y qui sont extrémités terminales d'un chemin d'origine x

Remarque:
- On veut accéder très facilement aux successeurs d'un sommet
- Dangers d'un algorithme empirique:
    - non terminaison (circuit infini)
    - oubli d'un sommet

Proposition: Pour éviter les deux problèmes ci-dessus, on marque les sommets qui sont visités les structures de données utilisées:
1- Un tableau tel que :

Marque[x] = 0, si le sommet x n'a pas été visité

Marque[x] = 1, si le sommet x a été visité

2- Une file, qui à tout moment contiendra les sommets qui ont été visités et dont on n'a pas visités tous les successeurs

Présentation globale de l'algorithme:
<u>Initialisation</u>

```
    Créer(F);
    Pour i <- 1 à n faire
        Marque[x_i] <- 0;
    Ajouter(x, F);
    Marque[x] <- 1;
```

<u>Traitement</u>

```
    Tant que Vide(F) = Faux faire
        Début
```

y <- Valeur(F);
Supprimer(F); {car stocker dans y donc oklm}
Pour tout successeur z de y faire
Si Marque[z] = 0 alors
Début
Ajouter(z, F);
Marque[z] <- 1;
Fin;
Fin;

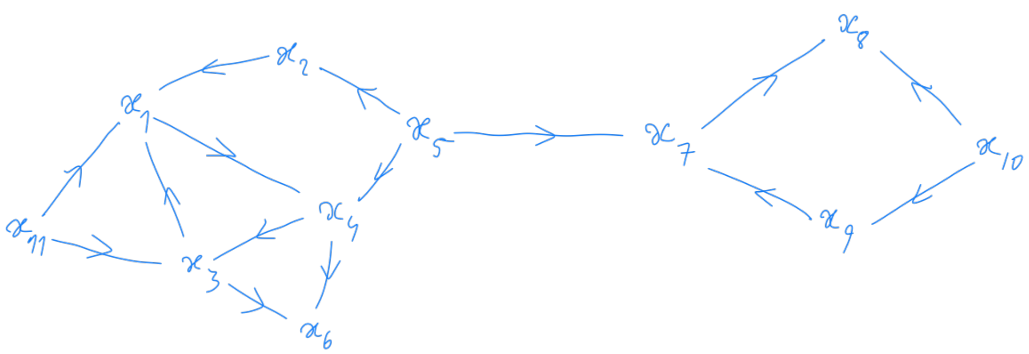
On pourrait renvoyer le tableau Marque[]

Complexité: N+M
N: nombre de sommets
M: nombre d'arcs

<u>Sortie des résultats</u>
Pour i <- 1 à n faire
Si Marque[x_i] = 1 alors
Écrire (x_i);

Remarque: Le temps de calcul est proportionnel Au nombre de sommets, la complexité est en $\theta(M+N)$

Exemple 18: Application de l'algorithme à l'exemple ci-dessous
Sommet de départ x_4



Marque	1	1	1	1	1	1	1	1	0	0	0
File	x_4	x_3	x_6	x_1	x_5	x_2	x_7	x_8			

5- Algorithme De Trémaux (parcours en profondeur)

Présentation globale

Algorithme de Trémaux
Trémaux(y)
Début
Pour tout successeur z de y faire
Si Marque[z] = 0 alors
Marque[z] <- 1;
Trémaux(z);
Fin

<u>Initialisation</u>

Pour i <- 1 à n faire

Marque[x_i] <- 0;

<u>Traitement</u>

Trémaux(x);

<u>Sortie des résultats</u>

Pour i <- 1 à n faire

Si Marque[x_i] = 1 alors

Écrire (x_i);

Exemple 19: application de Trémaux avec x_0 à

Ordre d'appel

T(x_O)

T(x_1)

T(x_3)

T(x_5)

T(x_4)

T(x_2)

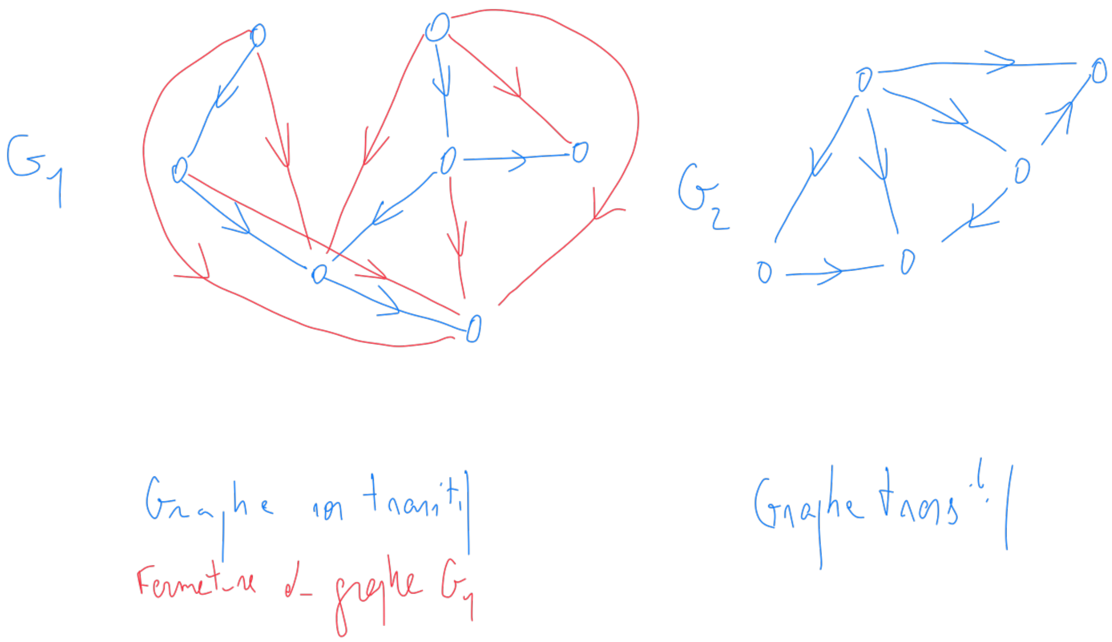
Exercice: Faire Trémaux en itératif

6- Construction de la fermeture transitive d'un graphe

Définition: Un graphe G est transitif si pour tout chemin C de G, il existe un arc $u \in G$ ayant même origine et même terminaison que C

Exemple 20:

Exemple 21: Fermeture des graphes de l'exemple 20



Définition: La fermeture d'un graphe $G = (X, U)$

1- Le plus petit graphe (X, U') tel que $U \subseteq U'$ et (X, U) est transitif

2- Le graphe obtenu à partir de (X, U) en ajoutant des arcs entre les sommets reliés par un chemin

Premier algorithme pour la construction de la fermeture transitive:

```
Pour tout sommet x faire
        début
                Trémaux(x)
                Ajouter un arc entre x et les sommets marqué, s'il n'existe pas déjà
        Fin;
```

Second algorithme pour la construction de la fermeture transitive:

Soit un graphe $G = (X, U)$ tel que $\|X\| = n$, et soit M sa matrice d'adjacence
M^P représente la matrice d'adjacence d'un graphe où un lien entre i et j signifie « Il existe dans G un chemin de longueur P allant de i vers j »
La matrice d'adjacence M^* de la fermeture transitive de G est donné par:

$$M^* = \sum_{i=1}^{n} M^i$$

Troisième algorithme pour la construction de la fermeture transitive:

Soit un graphe $G = (X, U)$ tel que $\|X\| = n$
On considère l'opération θ, qui, appliquée à G, fournit le graphe $\theta_r(G)$
Par définition, $\theta_n(G)$ comporte les arcs de U ainsi que tout arc (i, j) (si celui-ci n'appartient pas à G) tel qu'existe dans G les arcs (i, r) et (r, j)
Autrement dit, θ, enrichit G des arcs (i, j) (si celui-ci n'existait pas) tel qu'il existe un chemin de longueur 2 de i vers j en passant par r.
Opération θ_r:

```
Pour tout prédécesseur i de r faire
        Pour tout successeur j de r faire
                Ajouter l'arc de i à j
```

Algorithme de Roy-Warshall

Appliquer θ_1 à G, θ_2 à $\theta_1(G)$ puis θ_3 à $\theta_2(G)$, $\ldots \theta_n$ à $\theta_{n-1}(G)$
Le dernier graphe obtenu est la fermeture transitive de G

7- Forte connexité d'un graphe

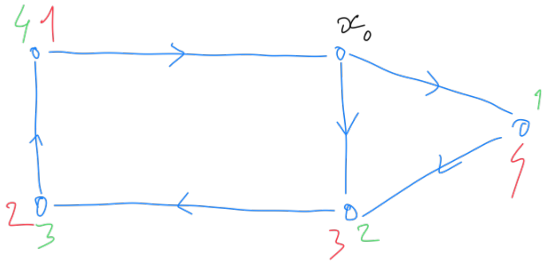
Soit un graphe $G = (X, U)$, on recherche la composante fortement connexe d'un sommet.

Présentation globale

Début
- Construction de G', le graphe symétrique de G (obtenu en inversant les arcs de G)
- Parcours du graphe G (en largeur ou Trémaux) à partir du sommet x_0 (donné)
- Parcours du graphe G' (en largeur ou Trémaux) à partir du sommet x_0
- Les sommets marqués deux fois constituent le composant fortement connexe du sommet x_0

Fin

Exemple 22:



8- Recherche de chemins existants

Le problème consiste à déterminer tous les couples de noeuds (i, j) tels qu'il existe un chemin de longueur k donnée reliant i et j
On peut partir de la matrice d'adjacente, qui n'est autre que la table des chemins de longueur 1
Soit M_1 cette matrice. On la considère comme une matrice booléenne dont chaque m_{ij} est respectivement vrai ou faux suivant que l'arc joignant le noeud origine i au noeud terminal j existe ou non

Algorithme:
M^p donne l'existence des chemins de longueurs p (multiplication booléenne)

Algorithme:
$M \oplus M^2 \oplus M^3 \oplus \ldots \oplus M^n$ donne l'existence des chemins de longueur quelconque

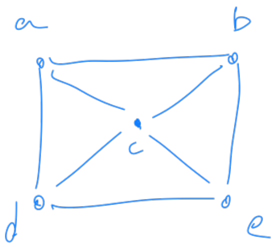
Algorithme:
Le nombre de chemin de longueur inférieur ou égale à p est donné par: $\sum_{i=1}^{p} M^i$

(Le prof à skip le 9- :/)

10- déterminer si un graphe est Eulérien

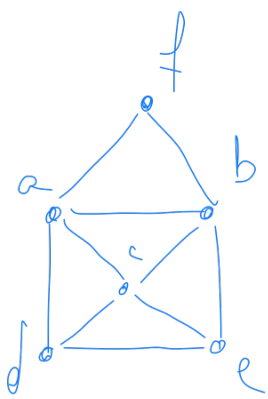
Un graphe Eulérien s'il possède un circuit (cycle) Eulérien
Un circuit (cycle, chemin, chaîne) est Eulérien, si la séquence d'arcs (arêtes) qui le compose comporte tous les arcs (arêtes) du graphe une et une seule fois

Exemple 25:



Ce graphe a-t-il un cycle Eulérien ?

Non

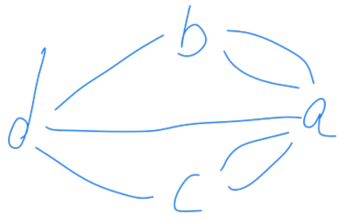


Ce graphe a-t-il un cycle Eulérien ? Non
A-t-il une chaîne Eulérienne ? Oui

$ecdebafbcad$

Question: Déterminer tous les cycles Eulérien du graphe

Exemple: Un problème célèbre de ce type est celui des sept ponts sur la Vistule à Koenigsberg: Comment passer sur tous les ponts sans passer deux fois par le même pont



Ce graphe a-t-il un cycle Eulérie ?

Réponse: NON

Une condition nécessaire et suffisante pour qu'il existe un circuit (cycle) Eulérien:

$$d^+_{(G)}(x) = d^-_{(G)}(x), \forall x \in X$$

Notation:

$G = (X, U)$

$d^+_G(x) = |\{u \in U / I(u) = x\}|$: nombre d'arc qui partent de u

$d^-_G(x) = |\{u \in U / T(u) = x\}|$: nombre d'arc qui arrivent à u

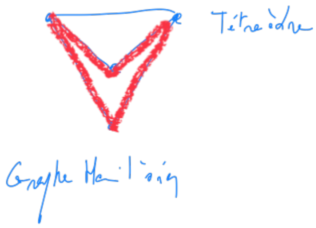
Remarque: Si le graphe n'est pas orienté, on considère la parité des arêtes des sommets

11- Déterminer si un graphe est Hamiltonien

Un graphe est Hamiltonien s'il possède un circuit (cycle) Hamiltonien

Un circuit (cycle, chemin, chaîne) est Hamiltonien, s'il passe une et une seule fois par chacun des sommet du graphe

Exemple 26:



Théorème (Camion 1959):
Tout 1-graphe complet fortement connexe admet un circuit Hamiltonien

p-graphe: Si le nombre d'arcs qui va d'un sommet x_i à un sommet x_j ne dépasse jamais un entier p

Graphe complet: $G = (X, U)$ est complet si $\forall x_i, x_j \in X; x_i \neq x_j$ on a:
$(x_i, x_j) \notin U \Rightarrow (x_j, x_i) \in U$

Exercice: Démontrer le théorème précédent

12- Les arbre en théorie des graphes

Définition: Un arbre est un graphe non orienté, connexe et sans cycle

Conséquence: Un arbre est un graphe simple sans boucle

Propriété: Un graphe est un arbre si et seulement si il est connexe et il contient $n-1$ arêtes (n étant le nombre de nœuds)

Exercice: Proposer une preuve de la propriété précédente

J'pose ça là, ça peut être utile
https://m.youtube.com/watch?v=l3sAb0ozDnY&feature=youtu.be

Propriété: Les conditions suivantes sont équivalentes

1- G est un arbre

2- G n'a pas de cycle et si on ajoute une arête il possède un cycle (G maximal sans cycle)

3- Pour tout couple de sommets, il existe un unique chemin qui les joint

4- G est minimal connexe (si on enlève une arête, il n'est plus connexe)

Exercice: Montrer que:

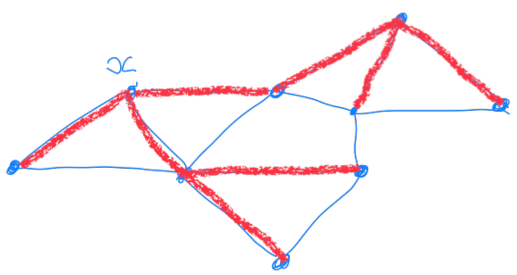
$$1- \Rightarrow 2-$$
$$2- \Rightarrow 3-$$
$$3- \Rightarrow 4-$$
$$4- \Rightarrow 1-$$

Théorème: Tout graphe connexe $G = (X, U)$ admet un graphe partiel $G' = (X, V)$ qui est un arbre

Algorithme (schéma) de recherche d'un graphe partiel qui soit un arbre

```
Début
        V <- Ø // initialisation
        Choisir un sommet x
        Marquer x par +
        Tant que x est marqué + mais pas marqué - faire
                Début
                        Si il existe y adjacent à x et non marqué + alors
                                Début
                                        Marquer y par +
                                        V <- V ∪ {x, y}
                                Fin
                        Sinon
                                Début
                                        Marquer x par -
                                        Si il existe y non marqué + mais pas marqué - alors
                                                x <- y
                                Fin;
                Fin;
Fin
```

Exemple 27:



13- Recherche d'un arc recouvrant de poids minimal

Graphe pondéré: On associe à chaque arête une valeur numérique qui représente le poids de l'arête

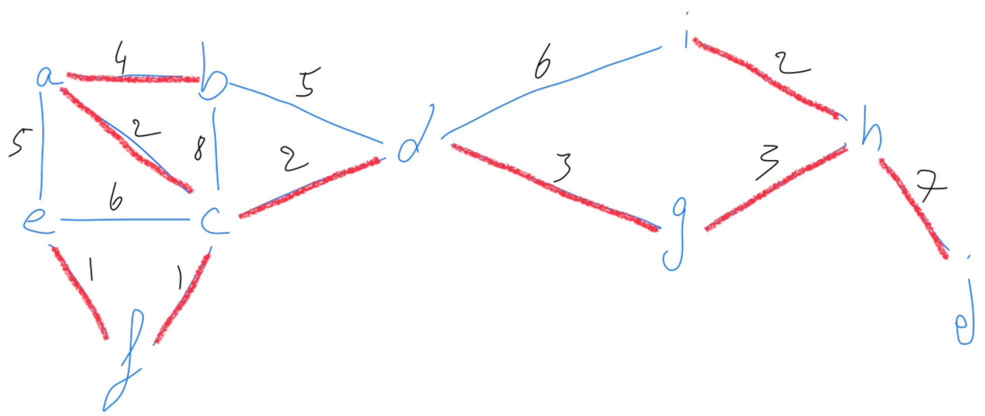
Arbre recouvrant d'un graphe: Étant donné un graphe non orienté formé de n nœuds, on appelle arbre recouvrant du graphe, toute composante connexe du graphe qui ne comporte aucun cycle et qui ne contienne que $n - 1$ arêtes

Arbre recouvrant de poids minimal graphe: C'est celui des arbres du graphe pondéré dont le poids a la valeur la plus petite

Algorithme de Kruskal

Ordonner les arêtes par valeur croissante
On obtient $u_1, u_2, ... u_n$ tel que valeur(u_1) ≤ valeur(u_2) ≤ … ≤ valeur(u_n)
V <- Ø // initialisation
Pour i -< 1 à n faire
    Si V ∪ $\{u_i\}$ n'a pas de cycle alors
        V <- V ∪ $\{u_i\}$

Exemple 28:



Écrire algorithme Kruskal avec PremSuc et TabSuc

14- Problèmes faciles et difficiles de la théorie des graphes

Problème facile: Il existe un algorithme qui répond à la question avec un nombre d'opérations élémentaires qui est une fonction de type polynomiale de la taille du graphe

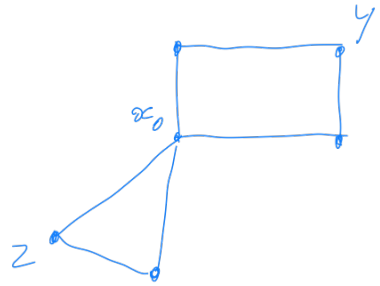
Fonction polynomiale: Qui ne croît pas trop vite

On ne peut pas espérer mieux que polynomiale (linéaire en n) $T = \mathcal{O}(n^3)$

Problème difficile: T est une fonction exponentielle de n, $T = e^n$

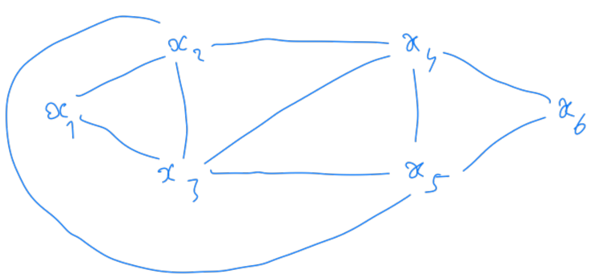
Exemple de problème facile:
1) Existence d'un chemin entre x et y
Savoir si graphe fortement connexe
Savoir si graphe est un arbre

Chemin longueur minimale
2) Trouver cycle Eulérien
3) Trouver arbre de poids minimal
4) Problème de connexité
Savoir s'il existe des points d'articulation (x_0 tel que $\exists$ y et z dont tous les chemins qui les joignent passent par x_0)(c'est un point tel que $G/\{x_0\}$ n'est pas connexe)
X_0 est un ensemble d'articulation si G/X_0 n'est pas connexe



5- Planarité
Étant donné un graphe peut-on le représenter dans le plan tel que ses arêtes ne se coupent pas ?

Exemple de problèmes difficiles:
1) Algorithme du voyageur de commerce, on se donne un graphe valué et on cherche un chemin qui passe par tous les sommets et dont la longueur est minimale. C'est un problème difficile car le nombre d'opérations n'est plus linéaire.
L'algorithme consiste à considérer tous les chemins, le coût est de l'ordre de $n!$
Pour $n = 30$ c'est impossible
2) Problème de coloration
Soit $G = (X, U)$ un graphe
On appelle coloration d'un graphe, une application: $X \longrightarrow \{1,2,...,k\}$ telle que si $(x, y) \in U$ alors $C(x) \neq C(y)$ (deux sommets adjacents n'ont pas la même couleur)
La question est de trouver le nombre minimum de couleurs
Question: Étant donné G, peut-on colorier ses sommets en 3 couleurs ?
Question: Étant donné G, trouver le plus petit k tel que G soit coloriable en k couleurs
Théorème: Tout graphe planaire est coloriable en 4 couleurs
3) Problème de cliques
Une clique dans un graphe c'est un ensemble de sommets
$x_1, x_2, \ldots, x_p$ tel que $\forall i, j, (x_i, x_j) \in U$



$\{x_1, x_2, x_3\}$ est une clique
$\{x_2, x_3, x_4, x_5\}$ est une clique

Déterminer la clique maximale est un problème difficile. On ne connaît pas l'algorithme dont le coût en temps est une fonction de type polynomial
4) Ensemble stable
$Y \subset X$ est stable. Y est un sous ensemble de sommets non reliés
$x_1, x_2, \ldots, x_p$ tel que $\forall i, j, (x_i, x_j) \notin U$
$\alpha(G)$: cardinale de plus grand ensemble stable de G

Problème: déterminer $\alpha(G)$
5) Problème du chemin Hamiltonien
Le chemin doit passer par tous les sommets une et une seule fois

Remarque: Tous ces problèmes sont équivalents entres eux