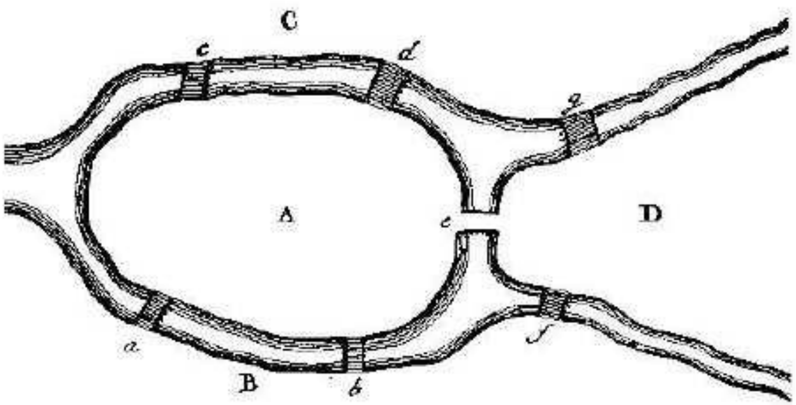
THÉORIE DES GRAPHES

- Concepts fondamentaux de la théorie des graphes
- Connexité dans un graphe
- Arbres et arborescences
- Le problème de recherche d'un plus court chemin
- Problème du flot
- Problème d'ordonnancement

UN BREF HISTORIQUE DE LA THÉORIE DES GRAPHES

- 1736 – Euler : les sept ponts de Königsberg



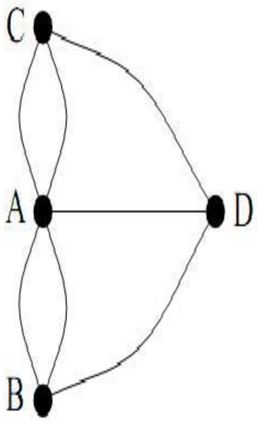


FIG. 2 – Graphe associé au problème des ponts de Königsberg

- Est-il possible, à partir d'une terre quelconque

A, B, C, ou D, de traverser chacun des ponts une et une seule fois et de revenir à son point de départ ?

Concepts fondamentaux de la théorie des graphes:

1-Définitions

Un Graphe

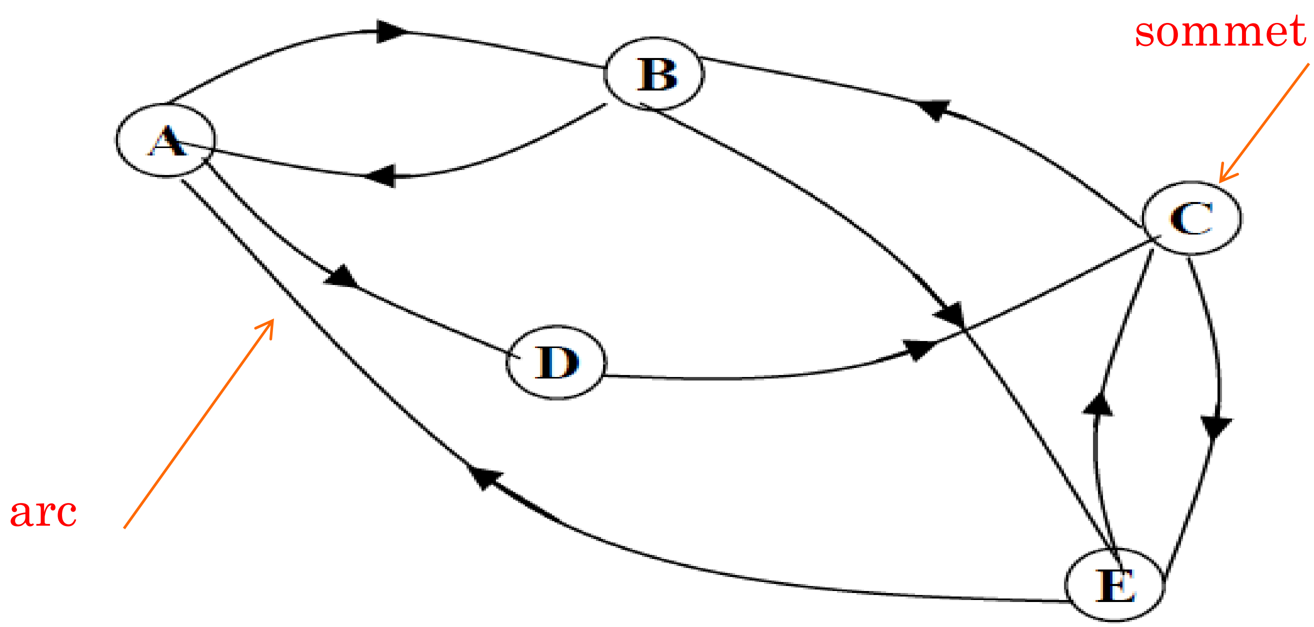
- Un graphe est un dessin géométrique défini par la donnée d'un ensemble de points (appelés **sommets** ou **noeuds**), reliés entre eux par un ensemble de **lignes** ou de **flèches** (appelées **arêtes** ou **arcs**). Chaque arête a pour extrémités deux points, éventuellement confondus.

- On appelle **ordre** d'un graphe le nombre de ses sommets
- On appelle **taille** d'un graphe le nombre de ses arêtes,

- Les graphes peuvent servir à représenter un grand nombre de situations courantes comme:
  - Les liens routiers
  - Les réseaux de communication
  - Les circuits électriques
  - Les liens entre diverses personnes ou entités administratives.

Exemple

- La figure suivante représente **un plan de circulation à sens unique d'une ville ou chaque** localité est représentée par un point appelé sommet et chaque route par un arc orienté indiquant le sens de la circulation.

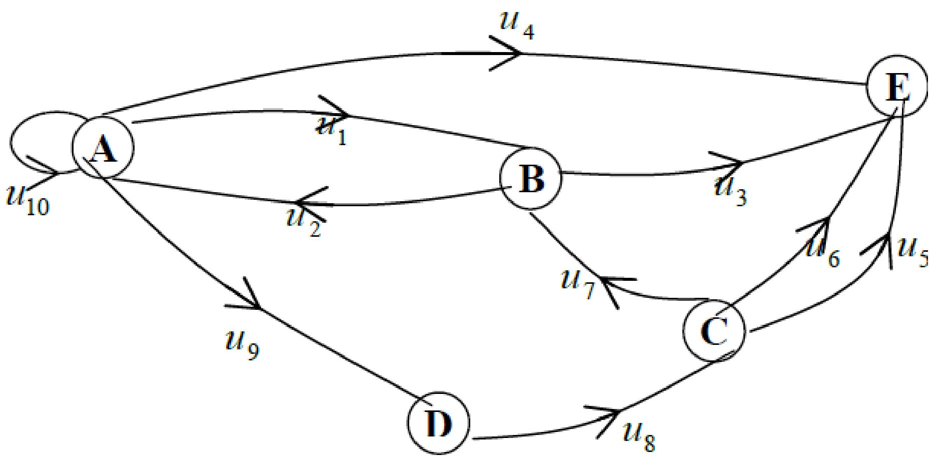


INCIDENCE ET *ADJACENCE*

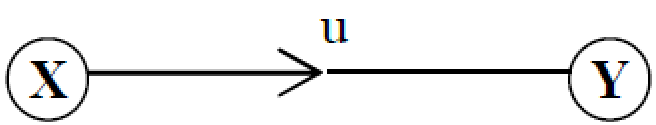
- Pour parler des relations qu'il y a entre deux sommets, deux arêtes, ou un sommet et une arête, on utilise les termes d'**incidence (*incidence*) et *d'adjacence* (*adjacency*).**
- On dit que deux sommets **a** et **b** sont adjacents si le graphe contient une arête *ab*. *On dit que deux* **arêtes sont adjacentes** s'il existe un sommet commun à ces deux arêtes.
- On dit qu'une arête est incidente à un sommet ou qu'un sommet est incident à une arête si le sommet est une des extrémités de l' arête

GRAPHE ORIENTÉ G(X,U)

- Un graphe orienté est un système formé d'un ensemble fini de **sommets** que l'on notera X $\{x_1, x_2, \ldots, x_n\}$ *et d'un* ensemble fini d'**arcs** reliant dans un ordre bien défini ces sommets, ou un certain nombres d'entre eux noté U $\{u_1, u_2, \ldots, u_m\}$



- On note un arc reliant un sommet x au sommet y dans un graphe G: par **u=(x,y)**



GRAPHE ORIENTÉ G=(X,U,I,T)

- Chaque arc du graphe G relie respectivement deux sommets, le sommet de départ qui représente l'extrémité initiale de l'arc et le sommet d'arrivée qui représente l'extrémité terminale

❖ **Autrement dit:**

Un graphe orienté est défini par le quadruplet: G=(X,U,I,T) où

- I est l'application extrémité initiale d'un arc définie par:

$$I: U \rightarrow X$$
$$(x,y) \rightarrow I(x,y)=x$$

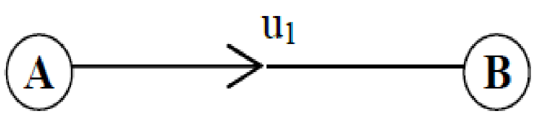
- T est l'application extrémité terminale d'un arc définie par:

$$T: U \rightarrow X$$
$$(x,y) \rightarrow T(x,y)=y$$

Exemple:

Soit u_1=(A,B) un arc de l'ensemble des arcs U du graphe G ci-dessus:

$I(u_1)$=A et $T(u_1)$=B



- **Remarque:**

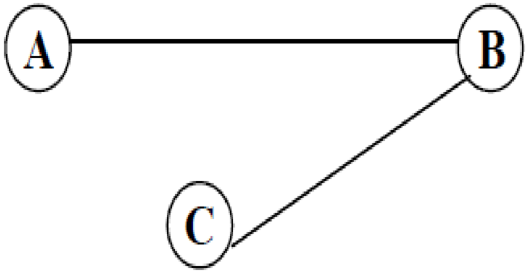
On appelle l'arc dont l'extrémité initiale est confondue avec l'extrémité terminale une **boucle** notée u(x,x)

- **Exemple:** Dans la figure u10=(A,A) / I(u10)=A et T(u10)=A . L'arc u10 est une boucle.

Graphe non orienté G=(X,E)

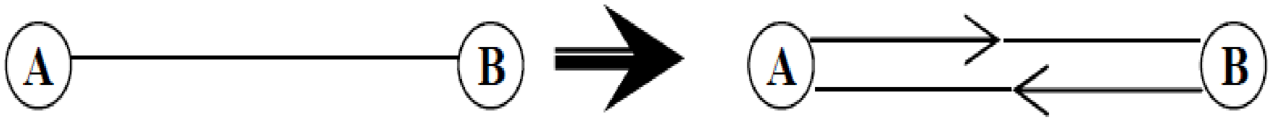
Si on définit une relation sur un ensemble où la notion d'ordre n'est pas importante, on représente ainsi la relation entre sommets par un arc non orienté appelé arête. On obtient alors un graphe non orienté, noté G=(X,E).

Exemple:



Remarque:

Une arête peut être transformée en deux arcs de sens déférents

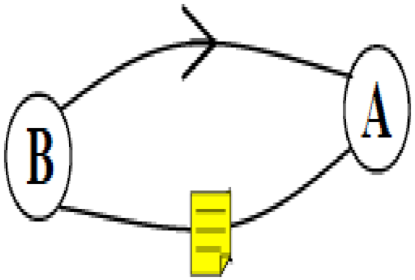


GRAPHE SIMPLE ET GRAPHE MULTIPLE:

Un graphe simple est un graphe sans boucles ni arcs (arête) multiples. Dans le cas contraire, on dira que le graphe est multiple.

Exemple:

Arêtes multiples

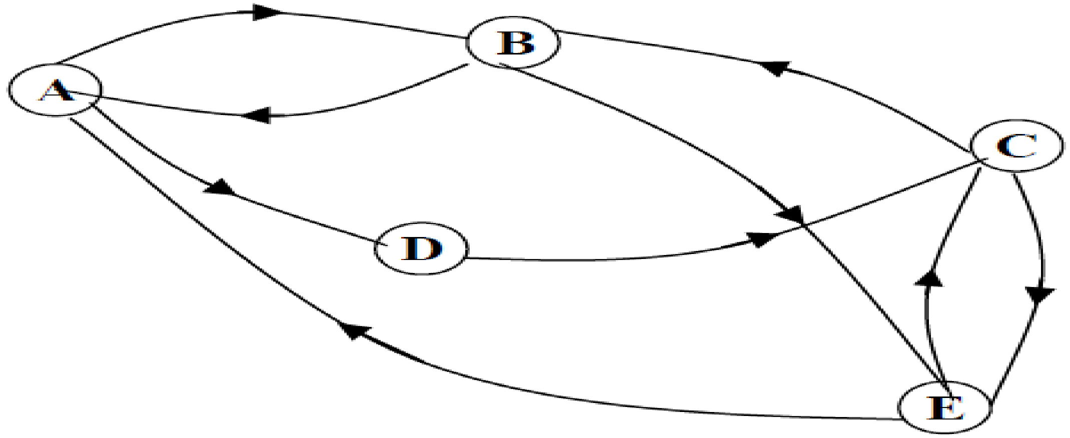


Arcs multiples

On définit ainsi, la multiplicité d'un graphe orienté multiple par le nombre maximum d'arcs ayant la même extrémité initiale et la même extrémité terminale. Soit p ce nombre, on dit G est un p-graphe.

L'ENSEMBLE DES PRÉDÉCESSEURS, SUCCESSEURS ET VOISINS D'UN SOMMET:

Considérons le graphe orienté suivant

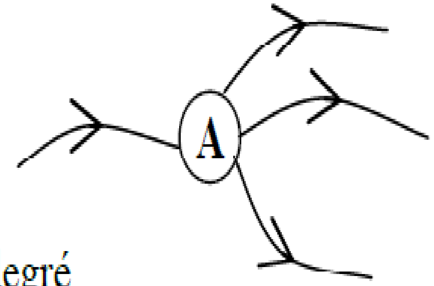


- De B et E on peut atteindre A par BA et EA. Donc, B et E forment l'ensemble des prédécesseurs de A, qu'on note $\Gamma^{-}(A)$.
- De A on peut atteindre B et D par AB et AD. Donc, B et D forment l'ensemble des successeurs de A, qu'on note $\Gamma^{+}(A)$.
- L'ensemble des voisins du sommet A est égale à la réunion de l'ensemble de ses prédécesseurs et de ses successeurs.

L'application Γ qui, à tout élément de X, fait correspondre une partie de X est appelée une application multivoque.

Degré d'un sommet

- le **degré** d'un sommet d'un graphe non orienté est le nombre de liens (arêtes ou arcs) reliant ce sommet, avec les boucles comptées deux fois
- Dans le cas d'un graphe orienté, on parle aussi du **demi degré intérieur (entrant)** d'un sommet s c'est-à-dire le nombre d'arcs dirigés vers le sommet s , et du **demi degré extérieur (sortant**)de ce sommet c'est-à-dire le nombre d'arcs sortant de s



- A est l'extrémité initiale de 3 arcs, on dit alors que le demi-degré extérieur de A est 3, on le note $d_G^+(A) = 3$.
- A est l'extrémité terminale d'un seul arc, on dit alors que le demi-degré intérieur de A est 1, on le note $d_G^-(A) = 1$.
- La somme du demi-degré intérieur et du demi-degré extérieur du A définit le degré de A, on le note $d(A) = 4$.

Remarque:

Degré de A égale à 0 → sommet isolé.

Degré de A égale à 1 → sommet pendant.

2- Structure d'un graphe:

SOUS GRAPHE ET GRAPHE PARTIEL

Considérons le réseau routier de l'Algérie G=(X,U) tel que:

X représente l'ensemble des villes d'Algérie et U représente l'ensemble des routes nationales et départementales algériennes.

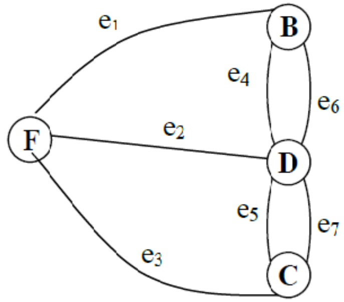
a) Soit $A \subset X$, l'ensemble des villes de la wilaya de Tizi Ouzou et U_A l'ensemble des routes reliant ces villes. On définit ainsi le graphe $G_A=(A,U_A)$ dit sous-graphe de G, représentant l'ensemble du réseau routier de la wilaya de Tizi Ouezou.

b) Soit $W \subset U$, l'ensemble des routes départementales Algériennes. On définit ainsi le graphe $G_W=(X,W)$, dit graphe partiel de G représentant les routes départementales Algériennes.

c) Soient U_A l'ensemble des routes reliant les villes de la wilaya de Tizi Ouezou (nationales et départementales) et W l'ensemble des routes départementales Algériennes. On définit ainsi le graphe $G_{AW} = (A, W \cap U_A)$, dit sous-graphe partiel de G représentant l'ensemble des routes départementales de la wilaya de Tizi Ouezou.

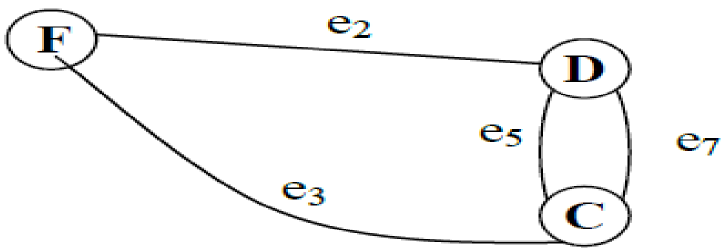
EXEMPLE

Soit le graphe G=(X,U): Soient $A=\{F,D,C\}$ et $W=\{e_1,e_2,e_5\}$

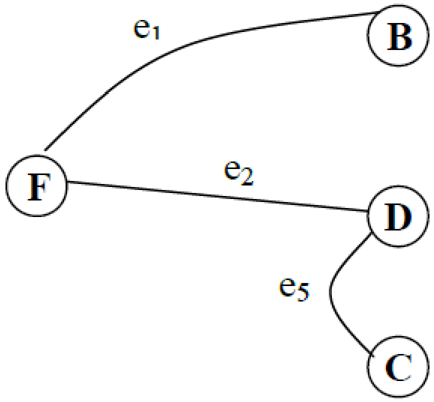


- **Le sous-graphe engendré par A**

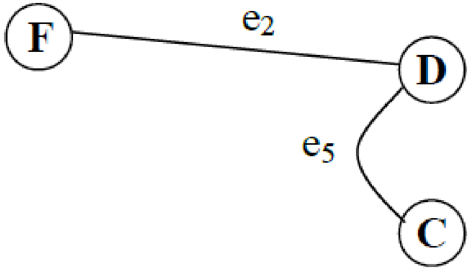
$G_A=(A,E_A)$, avec $E_A=\{e_2,e_3,e_5,e_7\}$



- **Le graphe partiel engendré par W est:**



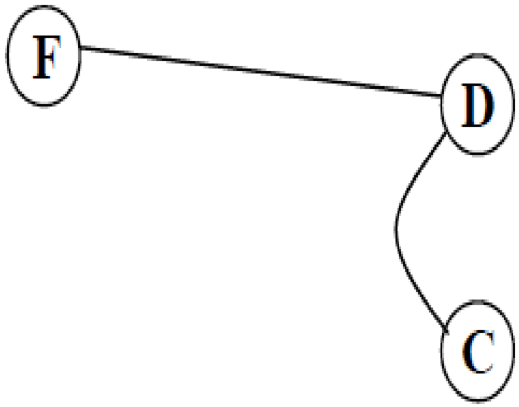
- **Le sous-graphe partiel engendré par A et W est:**



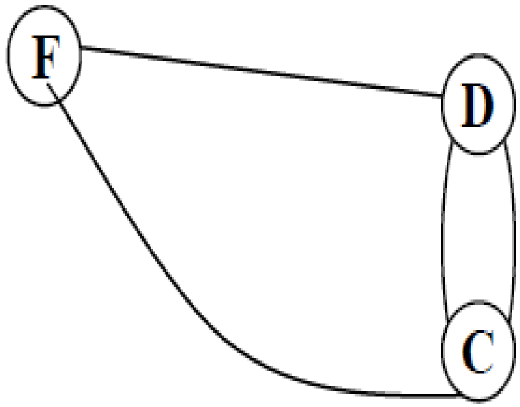
3-Les graphe particuliers

Graphe complet

- On appelle graphe complet un graphe dont tous les sommets sont adjacents (si, pour toute paire de sommets, il existe au moins un arc ou arrête).
- **Exemple**



G_1



G_2

Les sommets F et C dans G_1ne sont pas adjacents, le graphe est donc non complet.
Les sommets du graphe G_2 sont tous adjacents, d'où le graphe G_2 est complet.

- **Si un graphe est simple et complet d'ordre n , On le note K_n tel que n est le nombre des sommet**

Graphe complémentaire:

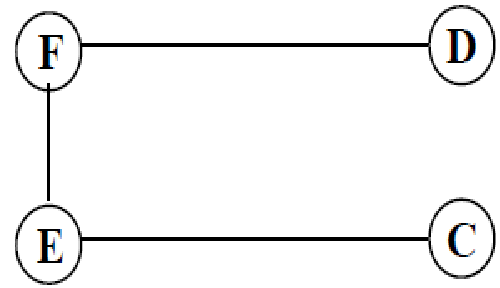
A un graphe simple G=(X,U), on peut définir un graphe complémentaire $\overline{G} = (X, \overline{U})$ comme suit: $u \in \overline{U} \Leftrightarrow u \notin U$.

C'est-à-dire: une arrête (arc) appartient au graphe complémentaire ($\overline{G}$)si elle n'appartient pas au graphe initiale G.

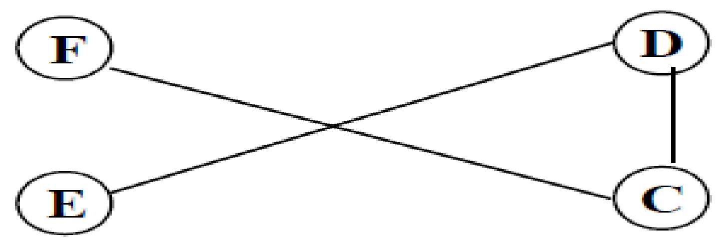
Exemple:

On considère le graphe simple suivant:

(G)



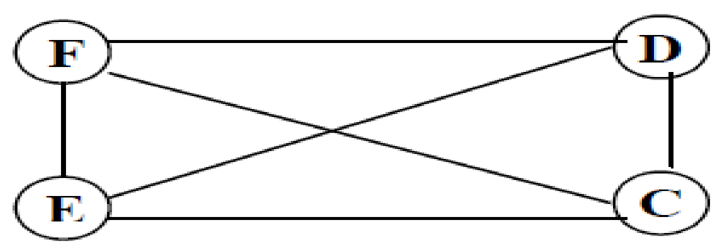
Son graphe complémentaire ($\overline{G}$) est:



Conséquence:

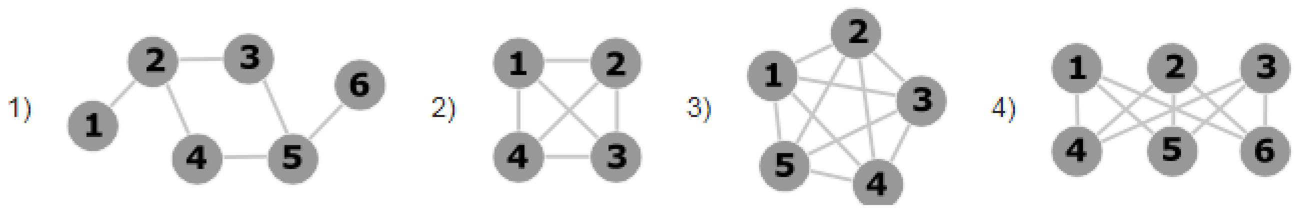
$G \cup \overline{G}$ est un graphe simple complet, donc un K_n.

$G \cup \overline{G} = K_4$



GRAPHE PLANAIRE:

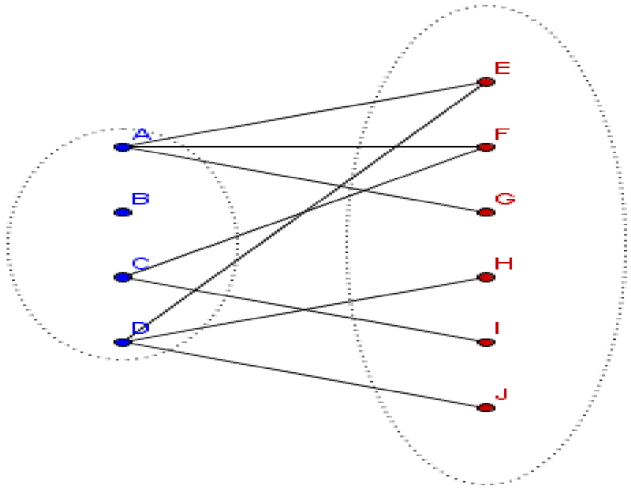
- Un graphe est dit planaire si on peut le dessiner sur un plan de telle façon que les arêtes ne se coupent pas, en dehors de leur extrémités.

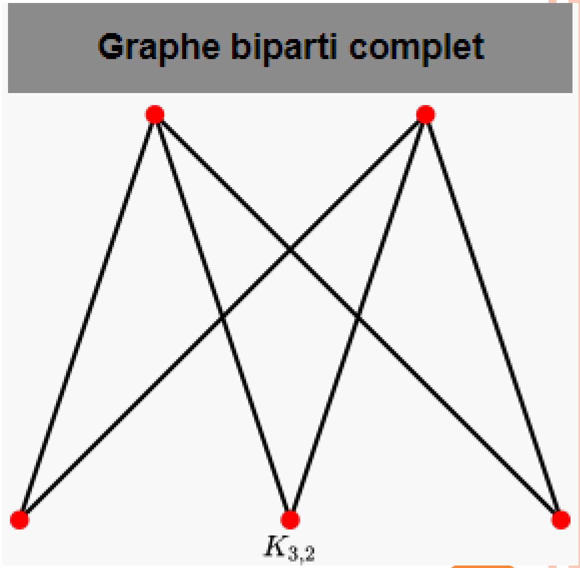


1.Ce graphe est clairement planaire, car il n'existe pas d'intersection entre deux arêtes.
2.C'est un graphe simple complet à quatre sommets (K4). Il est planaire : si on déplace le sommet *4* dans le triangle *1 2 3*, on constate qu'il n'y a plus d'intersection d'arêtes.
3.C'est un graphe complet à 5 sommets (K5). Il n'est pas planaire.
4.C'est un graphe biparti complet à 6 sommets, 3 d'entre eux se connectant aux trois autres (K3,3). Il n'est pas planaire.

Graphe biparti

- Un graphe non orienté est dit **biparti** si on peut partager son ensemble de sommets en deux parties A et B tels qu'il n'y ait aucune arête entre éléments de A et aucune arête entre éléments de B.





- un graphe est dit **biparti complet** s'il est biparti et contient le nombre maximal d'arêtes.

En d'autres termes, il existe une partition de son ensemble de sommets en deux sous-ensembles A et B telle que chaque sommet de A est relié à chaque sommet de B .

Il est Noté Km,n tel que m le nombre de sommet de l'ensemble A et n le nombre de sommet de l'ensemble B

4-Modes de représentation d'un graphe

LA REPRÉSENTATION MATRICIELLE:

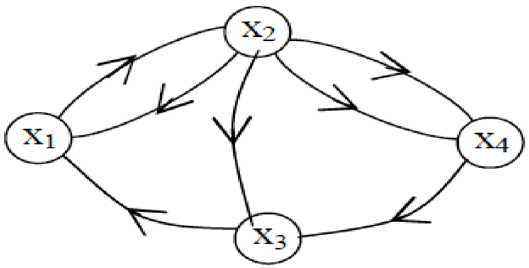
A un graphe G=(X,U) contenant n sommets et m arcs, on associera trois types de matrices:

1 La matrice d'adjacence:

La matrice d'adjacence du graphe G=(X,U) est une matrice n*n, ses éléments prennent deux valeurs 1 ou 0. Chaque ligne et chaque colonne correspondent à un sommet du graphe. Ainsi chaque élément de la matrice indique la relation qui existe entre deux sommets:

- 1 signifie que les deux sommets sont reliées par un arc orienté.
- 0 signifie que les deux sommets ne sont pas reliées par un arc.

Exemple:



(G)

La matrice d'adjacence de G est la suivante:

	x_1	x_2	x_3	x_4
x_1	0	1	0	0
x_2	1	0	1	1
x_3	1	0	0	0
x_4	0	0	1	0

- $u=(x_1,x_2)$ est arc du graphe G → $a_{12}=1$.
- pas d'arc ayant comme extrémité initiale x_1 et extrémité terminale x_3 → $a_{13}=0$.

Notation:

Les éléments de la matrice d'adjacence sont définis par:

$$a_{ij}=\begin{cases} 1 & \text{s'il existe un arc orienté } (x_i,x_j) \\ 0 & \text{sinon} \end{cases}$$

Dans le cas de graphes non orientés, la matrice est symétrique par rapport à sa diagonale descendante

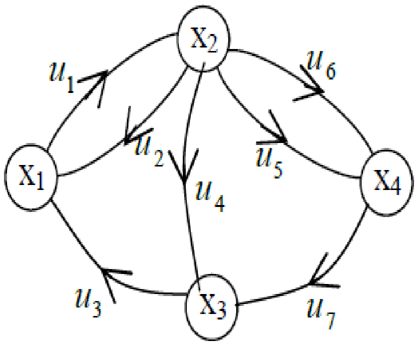
2- LA MATRICE D'INCIDENCE AUX ARCS:

- La matrice d'incidence aux arcs d'un graphe G=(X,U) est une matrice n*m, ses éléments prennent les valeur 1,0 ou -1.
- Chaque ligne de la matrice est associée à un sommet et chaque colonne à un arc. Chaque élément de la matrice indique la relation entre un sommet et un arc comme suit:
  - +1 signifie que le sommet est une extrémité initiale de l'arc.
  - -1 signifie que le sommet est une extrémité terminale de l'arc.
  - 0 signifie qu'il n'existe pas de relations entre le sommet et l'arc.
  - Une boucle a une double incidence sur un sommet, indiqué par 2 dans la matrice.

2- LA MATRICE D'INCIDENCE AUX ARCS:

Exemple:

Soit un graphe d'ordre 4, composé de 7 arcs.



(G)

La matrice d'incidence aux arcs de G est:

	u_1	u_2	u_3	u_4	u_5	u_6	u_7
x_1	+1	-1	-1	0	0	0	0
x_2	-1	+1	0	+1	+1	+1	0
x_3	0	0	+1	-1	0	0	-1
x_4	0	0	0	0	-1	-1	+1

Remarque:

Dans la matrice d'incidence on a:

- Le nombre de valeurs égale à +1 d'une ligne donne le degré extérieur du sommet correspondant.
- Le nombre de valeurs égale à -1 d'une ligne donne le degré intérieur du sommet correspondant.

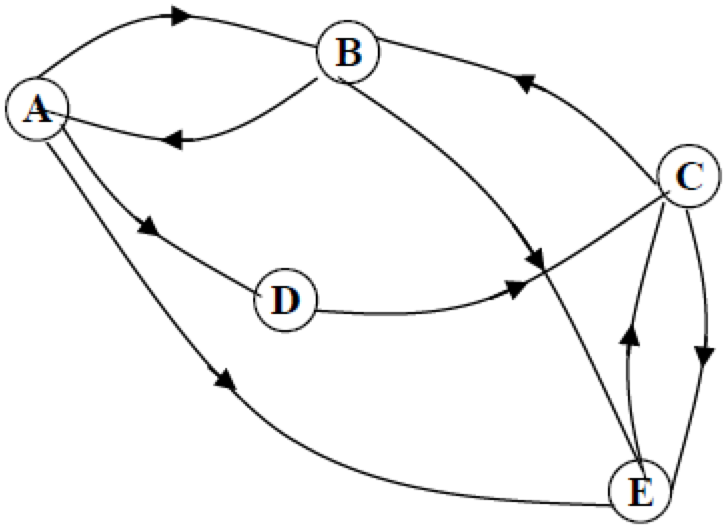
Connexité dans un graphe

1- Cheminements dans un graphe:

Les cheminements dans la théorie des graphes sont de quatre types: **la chaîne, le cycle, le chemin et le circuit.**

On définira ces notions dans le graphe G:

(G)



La chaîne:

Soit G=(X,U) un graphe.

Une chaîne joignant deux sommets x_0 et x_k dans un graphe G est une suite de sommets reliés par des arêtes tels que, deux sommets successifs ont une arête commune. On la note: $(x_0,x_1,\ldots,x_k)$.

Exemple:

Dans le graphe G, (A,B,C,D) est une chaîne.

Une chaîne est dite simple si on passe une seule fois par ses arcs.

Exemple:

Dans le graphe G,(A,B,E) ,(A,E,B,A,E) sont des chaînes joignant les sommets A et E, la seconde n'est pas simple car l'arc (AE) est parcouru deux fois.

Le chemin:

Soit G=(X,U) un graphe.

Un chemin du sommet x_0 à x_k dans un graphe G, est une suite de sommets reliés successivement par des arcs orientés dans le même sens. On le note: $(x_0,x_1,\ldots,x_k)$.

Exemple:

Dans le graphe G, (A,D,C,E) est un chemin joignant A à E.

Un chemin est dit simple si on passe une seule fois par ses arcs.

Exemple:

Dans le graphe G,(A,D,C,E) ,(A,D,C,E,C,E) sont des chemins joignant les sommets A et E, le second chemin n'est pas simple car l'arc (CE) est parcouru deux fois.

LE CYCLE:

- **Chaîne fermée / Chemin fermé**

Un chemin (resp. chaine) dont les extremites sont confondues est dit chemin ferme (resp. chaine fermee).

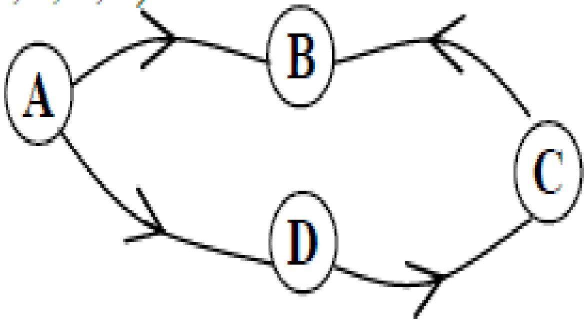
- **Cycle**

Un cycle est une chaîne simple dont les deux extrémités coïncident. On le note $(x_0, x_1, \ldots, x_k = x_0)$.

Exemple:

Dans le graphe G, la suite de sommets suivante (A,B,C,D,A)

Remarque: une boucle est un cycle particulier.

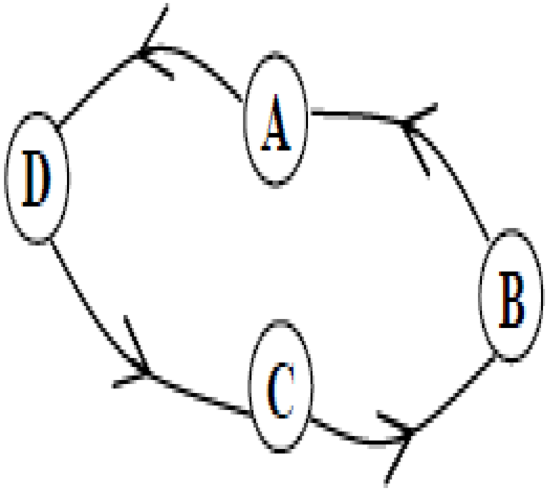


Le circuit:

Simple

Un circuit est un chemin dont les deux extrémités sont confondues; on le note $(x_0,x_1,\ldots,x_k=x_0)$.

Dans le graphe G, la suite (A,D,C,B,A) est un circuit.



GRAPHE HAMILTONIEN ET EULÉRIEN

- Une chaîne (cycle-chemin-circuit) est dite **(hamiltonien)** si on passe une seule fois par tous ses sommets(tous les sommets sont différents).

-Graphe hamiltonien:Un graphe qui contient un cycle hamiltonien est appelé graphe hamiltonien.

-Graphe semi-hamiltonienUn graphe semi-hamiltonien est un graphe qui contient une chaîne hamiltonienne, mais pas de cycle hamiltonien.

- Une chaîne (cycle-chemin-circuit) simple est dite **eulérien**

si elle passe une fois et une seule par chaque arête du graphe.

- Le graphe *G* est un **graphe eulérien** si et seulement si il admet un cycle eulérien.

D'après le théorème d'Euler :

- Un graphe connexe admet une chaîne eulérienne si et seulement s'il possède zéro ou deux sommet(s) de degré impair.
- Un graphe connexe admet un cycle eulérien si et seulement s'il ne possède que des sommets de degré pair.

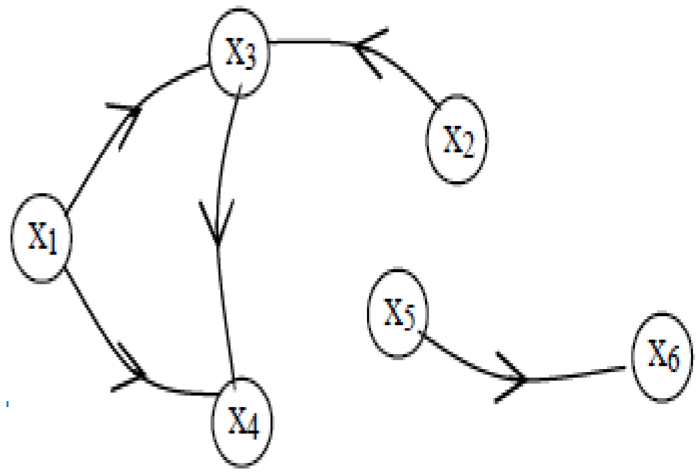
2- LA CONNEXITÉ:

LA NOTION DE CONNEXITÉ:

- On définit la connexité dans un graphe, par la relation entre deux sommets de la manière suivante:

deux sommets x et y ont une relation de connexité ↔ il existe une chaîne entre x et y ou bien x=y.

Exemple:
Soit le graphe (G) suivant:
- il existe une chaîne entre le sommet x1 et x2 notée C=(x1, x3, x2). Alors x1et x2 ont une relation de connexité.
- il n'existe pas de chaînes entre le sommet x1 et x5 ,alors x1 et x5 n'ont pas une relation de connexité.



Les composantes connexes:

- On appelle composante connexe un ensemble de sommets, qui ont deux à deux la relation de connexité, de plus tout sommet en dehors de la composante n'a pas de relation de connexité avec les sommets de cette composante.

Exemple:

Dans le graphe G de la figure précédente:

- les sommets x_1, x_2, x_3, x_4 ont deux à deux la relation de connexité, donc l'ensemble $\{x_1, x_2, x_3, x_4\}$ forme ainsi la 1ère composante connexe, on la note C_1.

- l'ensemble $\{x_5, x_6\}$ forme la 2ème composante connexe, on la note C_2.

On constate que les sommets de C_1 n'ont pas de relation de connexité avec les sommets de C_2.

Le graphe connexe:

- Un graphe G=(X,U) est dit graphe connexe si tous ses sommets ont deux à deux la relation de connexité; autrement dit, si G contient une seule composante connexe.
- Un graphe est connexe $\leftrightarrow$ il possède une seule composante connexe.

La recherche des composantes connexes:

- Les composantes connexes d'un graphe se déterminent en utilisant un algorithme de marquage simple.

Algorithme

Données: un graphe G=(X,U)

Résultat: le nombre k de composantes connexes de G ainsi que la liste $\{C_1, C_2, \ldots, C_k\}$de ses composantes connexes.

(0) Initialisation: k=0, W=X.

(1)

(1.1) Choisir un sommet de W et le marquer d'un signe (+), puis marquer tous ses voisins d'un (+). On continue cette procédure jusqu'à ce qu'on ne puisse plus marquer de sommets.

(1.2) Poser k=k+1 et C_k l'ensemble des sommets marqués.

(1.3) Retirer de W les sommets de C_k et poser $W=W-C_k$.

(1.4) On teste si W=Ø.

- Si oui terminer aller à (2).
- Si non aller à (1)

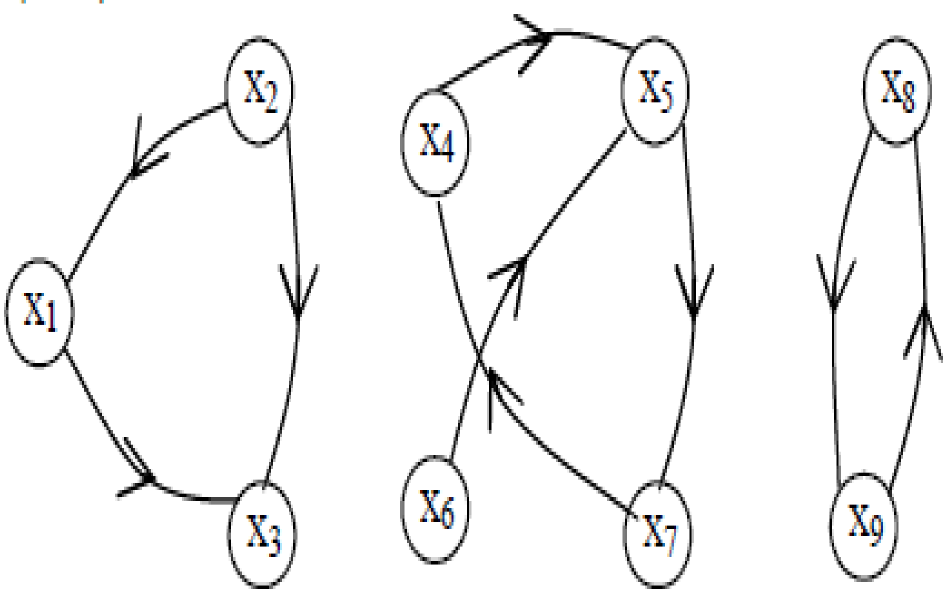
(2) Le nombre de composantes connexes de (G) est k.

Chaque ensemble C_i, i=1,..,k correspond aux sommets d'une composante connexe de (G).

La recherche des composantes connexes:

Application

Soit le graphe $G=(X,U)$ suivant:

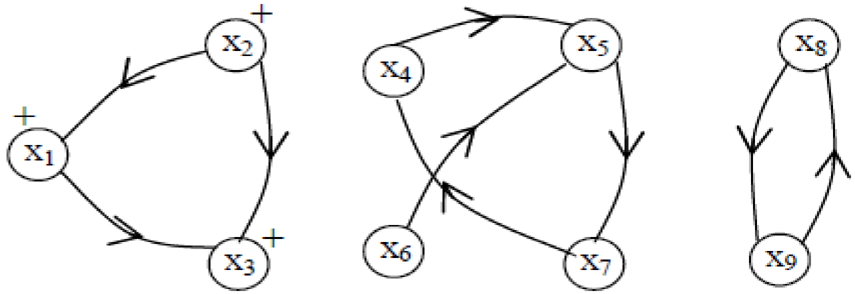


Le graphe (G) n'est pas connexe, car il n'existe pas de chaîne reliant les sommets x_3 et x_8.

On applique l'algorithme de marquage précédent pour déterminer les composantes connexes:

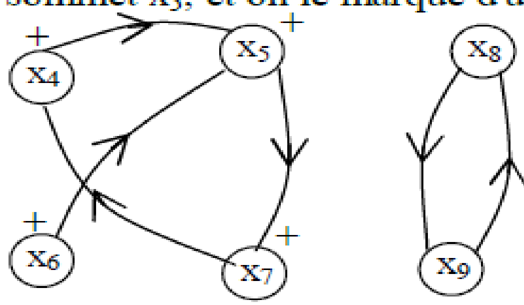
Initialisation: k=0, $W = X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$

Itération1: on choisir dans W le sommet x_2, et on le marque d'un signe (+), on marque ensuite ses voisins x_1 et x_3.



Soit $C_1 = \{x_1, x_2, x_3\}$ l'ensemble des sommets marqués.

Itération2: on choisir dans W le sommet x_5, et on le marque d'un signe (+), on marque ensuite ses voisins x_4,x_6 et x_7..



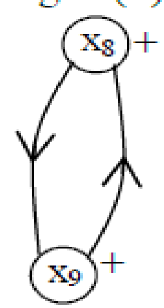
Soit $C_2 = \{x_4, x_5, x_6, x_7\}$ l'ensemble des sommets marqués.

On retire de W les sommets C_2, on obtient: $W = \{x_8, x_9\} \neq \Phi$

Itération3: on choisir dans W le sommet x_8, et on le marque d'un signe (+), on marque ensuite son voisin x_4,x_6 et x_7..

Soit $C_3 = \{x_8, x_9\}$ l'ensemble des sommets marqués.

Retirons de W les sommets C_3, on obtient: $W = \Phi$ terminer.



Le nombre de composantes connexes de (G) est 3, elles sont:

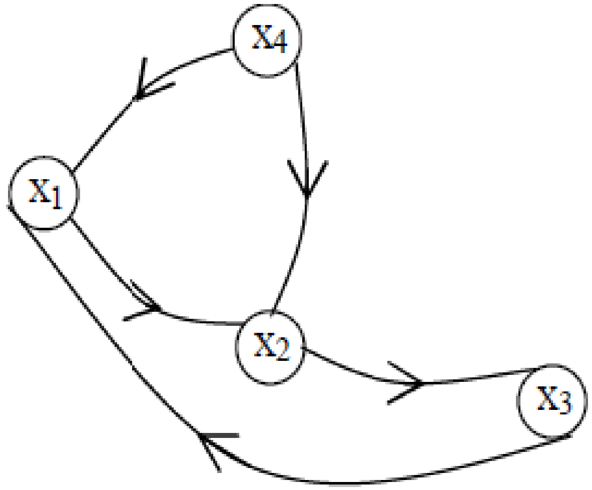
$C_1 = \{x_1, x_2, x_3\}$, $C_2 = \{x_4, x_5, x_6, x_7\}$ et $C_3 = \{x_8, x_9\}$

3-La forte connexité:

LA NOTION DE FORTE CONNEXITÉ:

- Un graphe orienté G=(X, U) est fortement connexe (f.c.) s'il existe entre chaque paire de sommets x et y ∈ X (x ≠ y) un chemin de x à y (xy) et un chemin de y à x (yx).

Exemple:
Soit le graphe G=(X,U) suivant:

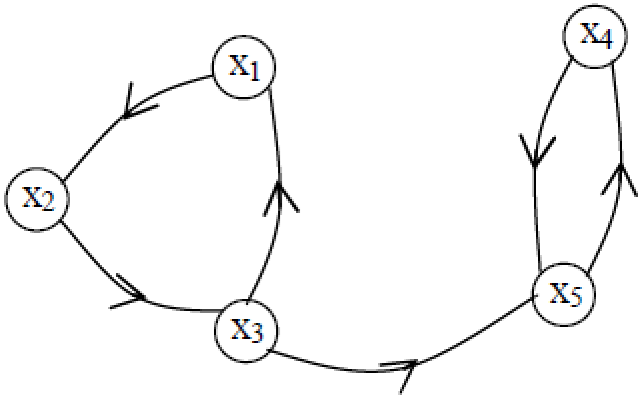


- On a un chemin reliant le sommet x_1 et x_3 et un chemin reliant le sommet x_3 au sommet x_1 alors x_1 et x_3 ont une relation de forte connexité.
- On a un chemin reliant le sommet x_4 à x_3, mais on n'a pas de chemin reliant le sommet x_3 au sommet x_4 alors x_4 et x_3 n'ont pas de relation de forte connexité.

Les composantes fortement connexes:

- On appelle composante fortement connexe un ensemble de sommets, qui onT deux à deux la relation de forte connexité, de plus tout sommet en dehors de la composante n'a pas de relation de forte connexité avec aucun élément de cette composante.

Exemple:
Soit le graphe G=(X,U) suivant:



- Les sommets x_1,x_2,x_3 ont deux à deux la relation de forte connexité, donc l'ensemble $\{x_1, x_2, x_3\}$ forme ainsi la 1$^{\text{ère}}$ composante fortement connexe, on la note C_1.
- L'ensemble $\{x_4, x_5\}$ forme la composante fortement connexe, on la note C_2.

Le graphe réduit:

- On appelle "graphe réduit" du graphe G=(X,U), le graphe Gr=(Xr,Ur) dont:

- Les sommets sont représentés par les composantes fortement connexes Ci du graphe G.

- Les arcs (x,y) dans le graphe G avec le sommet x appartenant à Ci et le sommet y appartenant à Cj, alors il existera un arc (Ci,Cj) dans le graphe réduit Gr.

Exemple:

Pour le graphe précédant, on a:

Le graphe fortement connexe:

- Un graphe G est dit fortement connexe si tous ses sommets ont deux à deux la relation de forte connexité, autrement dit si G contient une seule composante fortement connexe.
- **Algorithme**

Données: un graphe orienté G=(X,U).

Résultat: le nombre k de composantes fortement connexes de G ainsi que la liste $\{C_1, C_2, \ldots, C_k\}$ de ses composantes fortement connexes.

(0) Initialisation: k=0, W=X.

(1)

(1.1) Choisir un sommet de W et le marquer d'un signe (+) et (-).

(1.2) Marquer tous les successeurs directs et indirects de x avec (+).

(1.3) Marquer tous les prédécesseurs directs et indirects de x avec (-).

(1.4) Poser k=k+1 et C_k l'ensemble des sommets marqués avec (+) et (-).

(1.5) Retirer de W les sommets de C_k et effacer toutes les marques; on pose W=W-C_k.

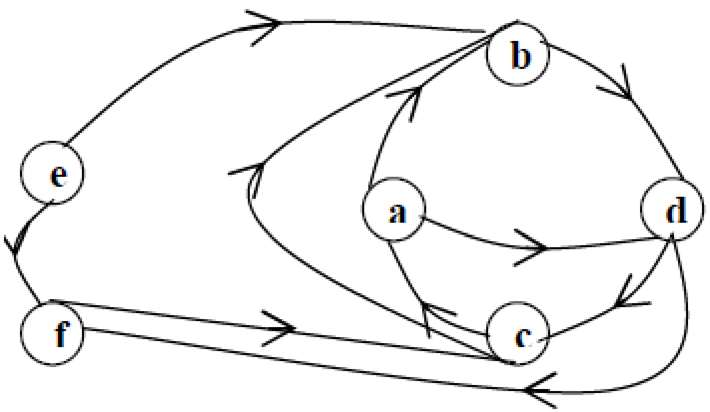
(1.6) On teste si W=Ø.

- Si oui terminer aller à (2).
- Si non aller à (1)

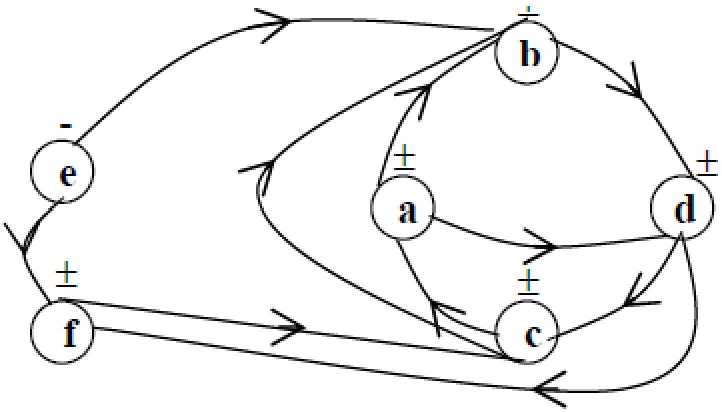
(2) Le nombre de composantes fortement connexes de (G) est k. Chaque ensemble C_i, i=1,..,k correspond aux sommets d'une composante fortement connexe de (G).

APPLICATION:

Soit le graphe G=(X,U) suivant:



itération1



$C_1 = \{a, b, c, d, f\}$ l'ensemble des sommets marqués à la fois par (+) et (-).

Itération2: on marque le sommet e d'un signe (+) et (-), on constate qu'il n y a pas d'autre sommets à marquer. Alors $C_2 = \{e\}$ l'ensemble des sommets marqués à la fois par (+) et (-).

On retire de W les sommets de C_2 on obtient: $W = \Phi$, terminer.

Le nombre de composantes fortement connexes de (G) est 2, elles sont: $C_1 = \{a, b, c, d, f\}$ et $C_2 = \{e\}$.

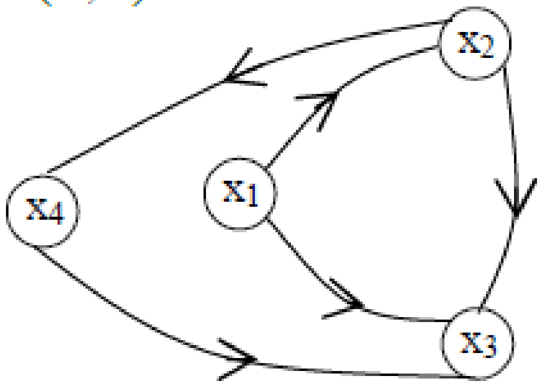
Au graphe (G), on fait correspond un graphe réduit noté G_r représenté comme suit:

4- La mise en ordre d'un graphe connexe

La mise en ordre d'un graphe connexe (l'ordonnancement d'un graphe)

- *Le principe:* Ordonner un graphe revient à disposer dans un certain ordre ses sommets tels que les arcs soient dans le même sens. On définit ainsi les différents niveaux des sommets du graphe.

Exemple: Soit le graphe G=(X,U) suivant:

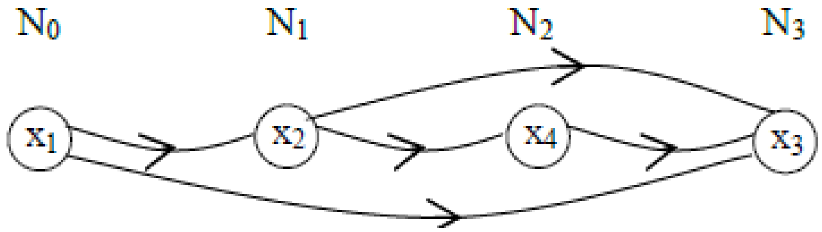


La mise en ordre d'un graphe connexe (l'ordonnancement d'un graphe)

Les niveaux des sommets du graphe sont définis comme suit:

- Le niveau nul, noté N_0 détermine les sommets du graphe n'ayant pas de prédécesseurs $(\Gamma_G^-(x) = \Phi)$. Dans le graphe G le sommet x_1 n'a pas de prédécesseurs d'où $N_0 = \{x_1\}$.
- Le premier niveau noté N_1 définit les sommets du graphe dont tous les prédécesseurs appartiennent à N_0. Dans le graphe G le sommet x_2 admet comme prédécesseurs le sommet x_1 qui appartient au niveau N_0 d'où $N_1 = \{x_2\}$.
- Le deuxième niveau noté N_2 définit les sommets du graphe dont tous les prédécesseurs appartiennent à $N_0 \cup N_1$. Dans le graphe G le sommet x_4 admet comme prédécesseurs le sommet x_2 qui appartient à $N_0 \cup N_1$ d'où $N_2 = \{x_4\}$.
- Le troisième niveau noté N_3 définit les sommets du graphe dont tous les prédécesseurs appartiennent à $N_0 \cup N_1 \cup N_2$. Le sommet x_3 admet comme prédécesseurs le sommet x_1,x_2 et x_4 qui appartient à $N_0 \cup N_1 \cup N_2$ d'où $N_3 = \{x_3\}$.

Le graphe ordonné de G est:



La mise en ordre d'un graphe connexe (l'ordonnancement d'un graphe)

Algorithme

La mise en ordre d'un graphe connexe G=(X,U) ou l'ordonnancement d'un graphe se traduit par l'algorithme suivant:

Données: un graphe G=(X,U).

Résultat: les différents niveaux de sommets du graphe ainsi que le graphe ordonné de G.

(0) On détermine le dictionnaire des prédécesseurs du graphe G formé par le couple ($W, \Gamma_G^-(x)$).

(1) On repère dans le dictionnaire des prédécesseurs du graphe les sommets n'ayant pas de prédécesseurs ($\Gamma_G^-(x) = \Phi$).

(1.1) On pose N_0 l'ensemble des sommets du graphe n'ayant pas de prédécesseurs, on l'appelle niveau nul.

(1.2) On barre dans la colonne de $\Gamma_G^-(x)$ tous les sommets de niveaux nul N_0, on obtient une nouvelle colonne $\Gamma_{G1}^-(x)$, avec G_1 le sous-graphe engendré par X/N_0.

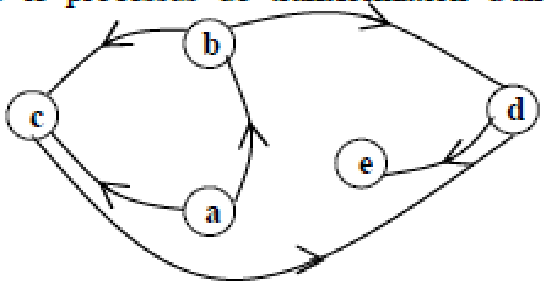
(2) On repère dans le nouvelle colonne $\Gamma_{G1}^-(x)$ les sommets n'ayant pas de prédécesseurs ($\Gamma_{G1}^-(x) = \Phi$).

(2.1) On pose N_1 l'ensemble des sommets du graphe n'ayant pas de prédécesseurs.

(2.2) On barre dans la colonne de $\Gamma_{G1}^-(x)$ tous les sommets de niveaux N_1, on obtient une nouvelle colonne $\Gamma_{G2}^-(x)$, avec G_2 le sous-graphe engendré par $X/N_0 \cup N_1$.

On continue le même procédé jusqu'à ce qu'on termine le graphe et on représente ainsi le graphe ordonné par niveaux de G.

Exemple: Soit le graphe G=(X,U) représentant le processus de transformation d'une matière première a dans un atelier.



1) Soit le dictionnaire des prédécesseurs du G:

Le sommet a n'a pas de prédécesseurs, il est donc de niveau nul $N_0 = \{a\}$

X	$\Gamma_G^-(x)$
a	Φ
b	~~a~~
c	~~a~~-b
d	c-b
e	d

on barre le sommet a de la colonne $\Gamma_G^-(x)$, on obtient la nouvelle colonne $\Gamma_{G1}^-(x)$.

2) Le sommet b n'a plus de prédécesseurs, il est donc de niveau un $N_1 = \{b\}$

X	$\Gamma_{G1}^-(x)$
a	-
b	Φ
c	~~b~~
d	c-~~b~~
e	d

4) le sommet d n'a plus de prédécesseurs, il est donc de niveau trois $N_3 = \{d\}$

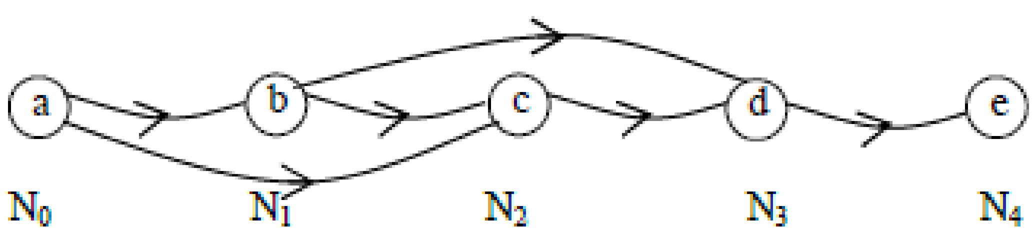
X	$\Gamma_{G3}^-(x)$
a	-
b	-
c	-
d	Φ
e	~~d~~

On barre le sommet d de la colonne $\Gamma_{G3}^-(x)$, on obtient le nouveau dictionnaire.

4) le sommet e n'a plus de prédécesseurs, il est donc de niveau quatre $N_4 = \{e\}$

X	$\Gamma_{G4}^-(x)$
a	-
b	-
c	-
d	-
e	Φ

On a examiné tous les sommets du graphe G, on a ainsi le graphe ordonné de G représenté ci dessous.



Remarque

A une étape donnée de l'ordonnancement d'un graphe, la définition des niveaux se bloque (i.e il n'existe pas de sommets n'ayant pas de prédécesseurs), donc la mise en ordre du graphe est impossible, on dit alors que le graphe possède un circuit.

PROPRIÉTÉS

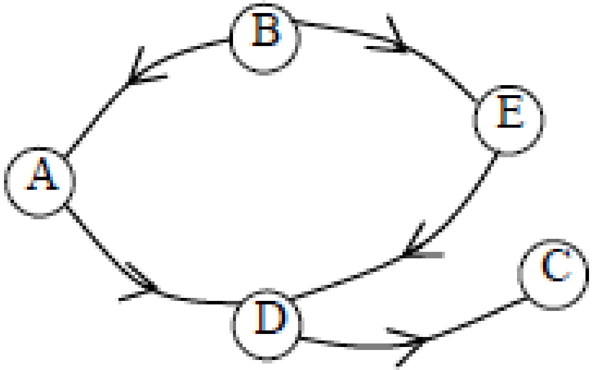
- *Propriété 1:*

  Soit n le nombre de sommets d'un graphe G=(X,U), et m le nombre de ses arcs.

  - Si G est connexe $\Rightarrow$ $m \geq n-1$
  - Si G est sans cycles $\Rightarrow$ $m \leq n-1$

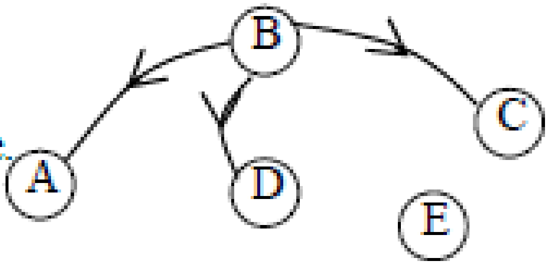
Exemple:

- Soit le graphe G=(X,U) suivant:



Le graphe G est connexe et contient un cycle C=(A,B,E,D).

- Soit le graphe G=(X,U) suivant:



Dans le graphe G, le nombre d'arcs (m=3) est inférieur au nombre de sommets (n-1=4), alors le graphe G n'est pas connexe.

Arbres et arborescences

1- Arbres et arborescences

Définition d'un arbre

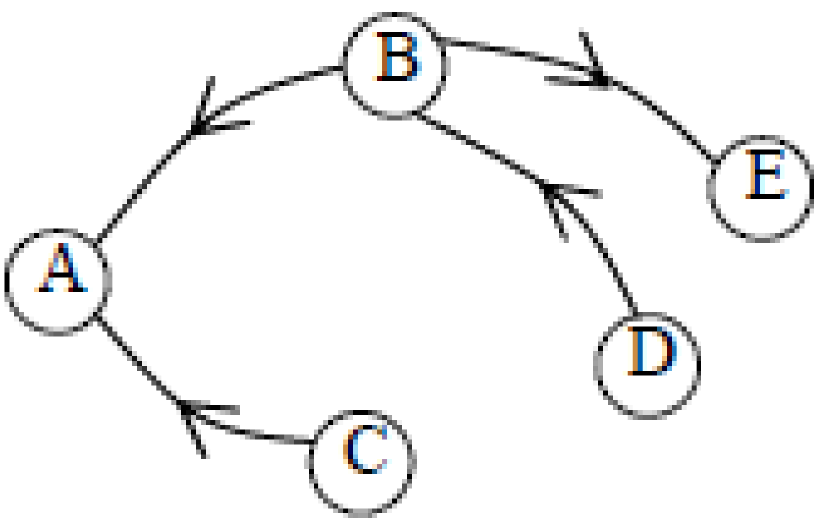
Un arbre, est par définition, un graphe connexe et sans cycl

- **Remarque:**

D'après la définition, un arbre est un graphe simple sans boucle, ayant (n-1) arcs.

- **Exemple:**

Le graphe G=(X,U) est un arbre.



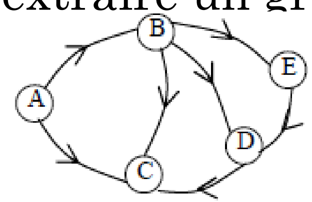
Propriété

Soit G=(X,U) un graphe sur n=|X|≥2 sommets. Les propriétés suivantes sont équivalentes et caractérisent un arbre:

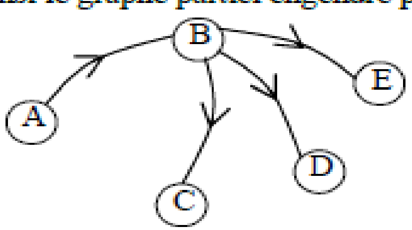
(i) G est connexe et sans cycle.
(ii) G est connexe et est minimal pour cette condition (si on supprime un arc de G, il ne sera plus connexe).
(iii)G est connexe et possède (n-1) arcs.
(iv) G est sans cycle et, est maximal pour cette propriété (si on ajoute un arc à G, il possédera un cycle).
(v) G est sans cycle et possède (n-1) arcs.
(vi) Entre chaque couple de sommets, il existe une et une seule chaîne les reliant.

Remarque:
D'un graphe connexe G=(X,U) on peut extraire un graphe partiel qui est un arbre.

Exemple:
Soit le graphe G=(X,U) suivant:



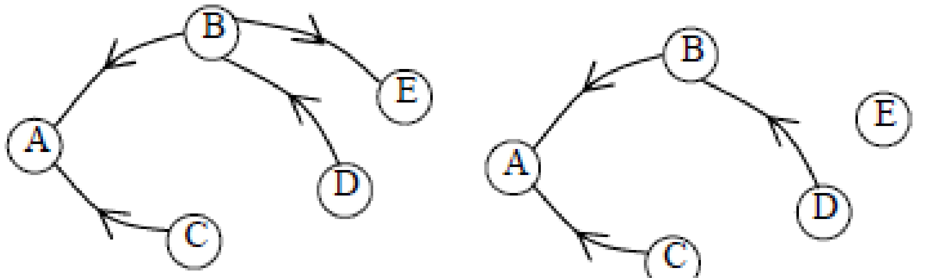
Soit $W = \{AB, BC, BE, BD\}; W \subset U$. On définit ainsi le graphe partiel engendré par W, G_W=(X,W) qui est un arbre.



DÉFINITION D'UNE FORÊT

Définition d'une forêt:
Une forêt est un graphe dont chaque composante connexe est un arbre. C'est-à-dire un graphe sans cycle.
Remarque:
- La forêt est obtenue si on relaxe (relâche) la contrainte de connexité dans un arbre, c'est-àdire, si on supprime un arc dans un arbre.
- Tout graphe partiel d'un arbre est une forêt.

Exemple:
Dans le graphe de la figure suivante, si on relâche la contrainte de connexité en supprimant par exemple l'arc BE, on obtient une forêt.



La forêt obtenue est un graphe partiel engendré par l'ensemble des arcs $W = \{BA, CA, DB\}$.

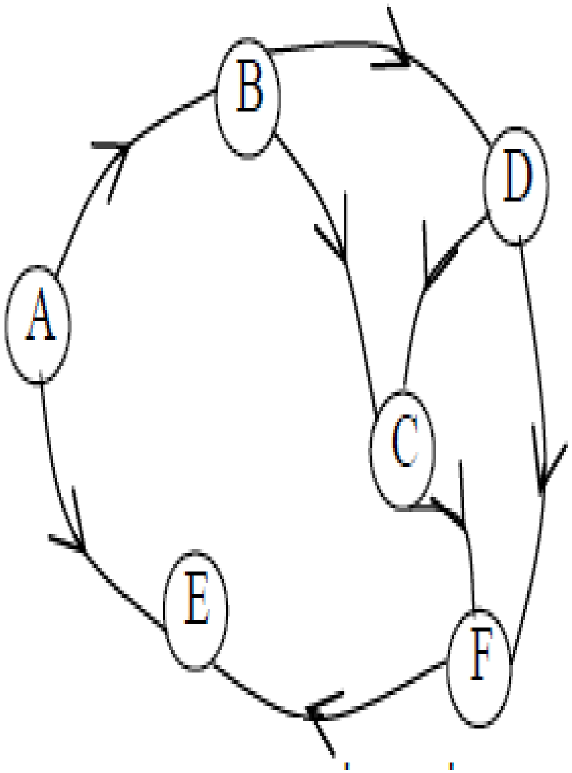
DÉFINITION D'UNE RACINE

Un sommet s d'un graphe G est une "racine" (resp, une "antiracine") s'il existe un chemin joignant s à chaque sommet du graphe G (resp. joignant chaque sommet de G à s) à l'exception du sommet lui-même. C'est-à-dire : $\forall x \in X - \{s\}$, il existe un chemin de s à x (resp de x à s).

Exemple:

Soit le graphe G=(X,U) suivant:

- Le sommet A est une racine du graphe G.
- Le sommet E est une antiracine du graphe G.



DÉFINITION D'UNE ARBORESCENCE:

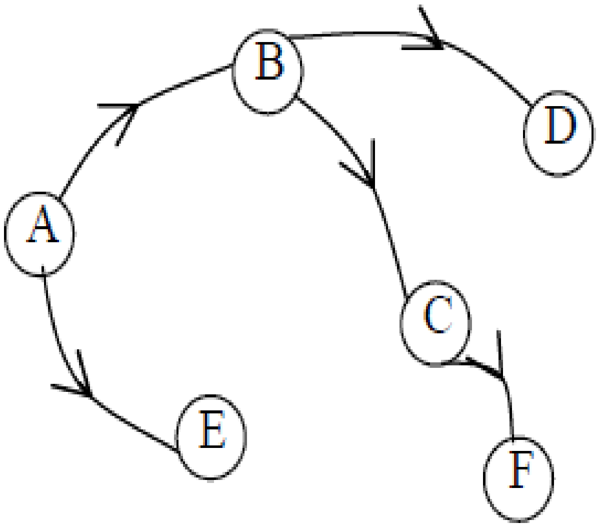
Un graphe G=(X,U), avec $|X| = n \geq 2$ sommets est une arborescence de racine s si:

- G est un arbre.
- s est une racine.

Exemple:

Soit W=(X,U) l'arbre suivant:

W est un arbre admettant le sommet A comme racine alors W est une arborescence.



Remarque:

Une arborescence est un arbre mais la réciproque est fausse.

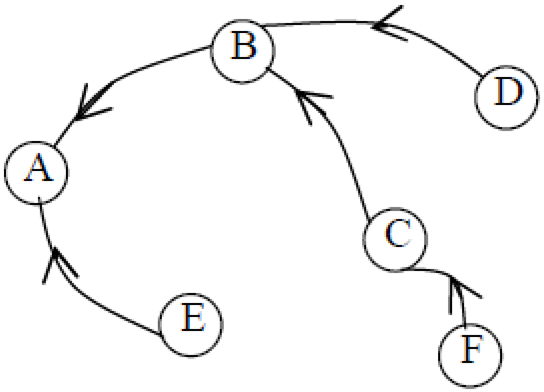
Définition anti-arborescence:

Un graphe G=(X,U) sur $n = |X| \geq 2$ sommets est une anti-arborescence admettant le sommet s comme antiracine si:

- G est un arbre.
- s est une antiracine de G.

Remarque:

Si on inverse le sens des arcs d'une arborescence, on obtient une anti-arborescence.



2-Le problème de recherche d'un arbre de poids minimum (minimum tree of pods)

Si on associe à chaque arc d'un graphe G=(X,U) une valeur (un poids). Le problème de l'arbre de coût minimum consiste à trouver un graphe partiel qui est un arbre, dont la somme des poids des arcs est minimale.

Exemple: minimiser le coût d'installation des lignes téléphoniques dans une localité peut être représenté comme un problème de recherche d'un arbre de coût minimum. En effet, on veut, relier tous les points de la localité sans avoir de lignes inutiles, d'où la recherche d'un arbre. Ensuite on veut avoir un coût d'installation minimum, alors on associera à chaque possibilité d'installation d'une ligne le coût nécessaire et on cherchera à minimiser le coût total de toute l'installation.

En 1956, J.B. Kruskal a donné un algorithme qui permet de résoudre un tel problème.

Algorithme de kruskal pour construire un arbre de poids minimum:

Le principe:
L'idée de l'algorithme de Kruskal est tout d'abord de **numéroter** les arcs par ordre des poids **croissants**. Ensuite de construire progressivement **l'arbre A** en rajoutant dans leurs ordre, les arcs un par un. Un arc est ajouté seulement si son adjonction à A **ne détermine pas de cycle**, c'est-à-dire si A ne perd pas sa notion d'arbre, sinon on passe à l'arc suivant dans l'ordre de la numérotation.

Algorithme de kruskal

Données: un graphe valué G=(X,U,c).

Résultat: ensemble d'arcs W

(0) Initialisation: numéroter les arcs de G dans l'ordre des poids croissants: $c(u_1) \leq c(u_2) \leq \ldots\ldots \leq c(u_m)$. Soit $W = \phi$; i=1.

(1) Si $(X, W \cup \{u_i\})$ contient un cycle aller en (3)

Sinon aller en (2)

(2) On pose $W := W \cup \{u_i\}$ aller (3)

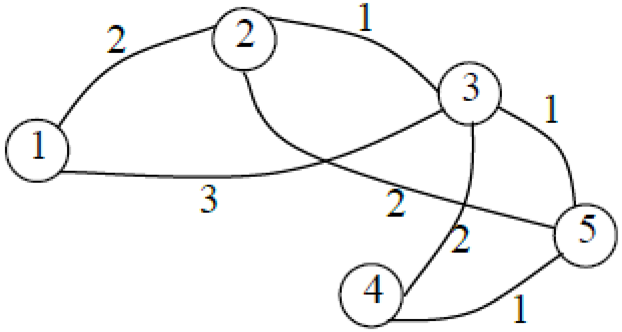
(3) Si i=m terminé, A=(X,W) est l'arbre de poids minimum $c(W) = \sum c(u_i)$ pour $u_i \in W$

Sinon i:=i+1 aller en (1)

L'algorithme s'arrête lorsque le nombre d'arcs retenus est égal à n-1.

APPLICATION:

Soit G=(X,U) un graphe connexe représentant le projet d'installation de lignes téléphoniques. Les poids représentent le coût d'installation des lignes.
On veut donner un plan d'installation minimisant le coût total de l'installation.



Initialisation: On ordonne les arêtes du graphe selon les poids croissants:

i	1	2	3	4	5	6	7
u_i	(2,3)	(3,5)	(4,5)	(3,4)	(2,5)	(1,2)	(1,3)
$c(u_i)$	1	1	1	2	2	2	3

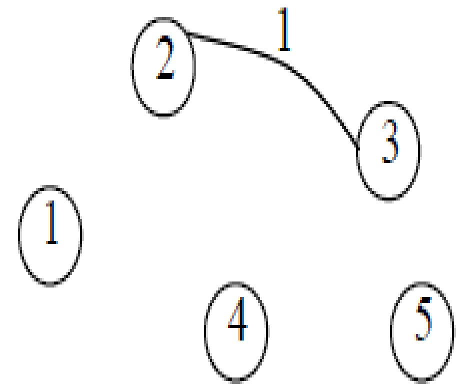
Soit $W = \phi$; i=1;m=7

- **Itération 1:**

On a : u_1=(2,3); Soit $W = W \cup \{u_1\} = \{(2,3)\}$

Le graphe (X,W) ci contre ne contient pas de cycle, on pose alors

$W = \{(2,3)\}$ $|W| = 1$. On a: i≠m alors on pose i=i+1=2.

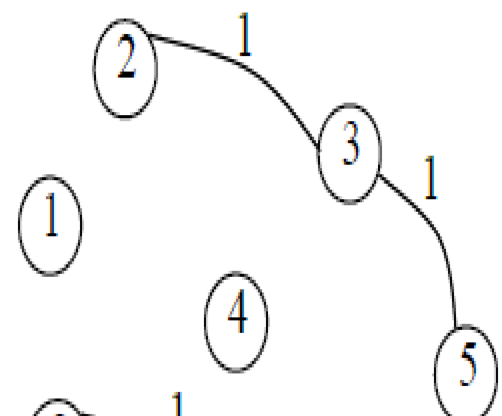


- **Itération 2:**

On a : u_2=(3,5); Soit $W = W \cup \{u_2\} = \{(2,3),(3,5)\}$

Le graphe (X,W) ci contre ne contient pas de cycle, on pose alors

$W = \{(2,3),(3,5)\}$ $|W| = 2$. On a: i≠m alors on pose i=i+1=3.

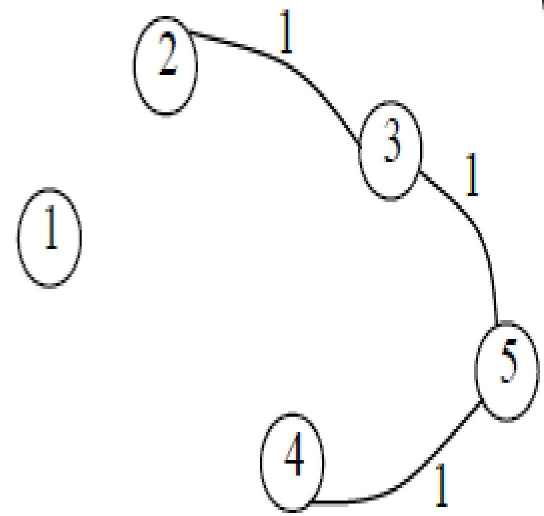


- **Itération 3:**

On a : u_3=(4,5); Soit $W = W \cup \{u_3\} = \{(2,3),(3,5),(4,5)\}$

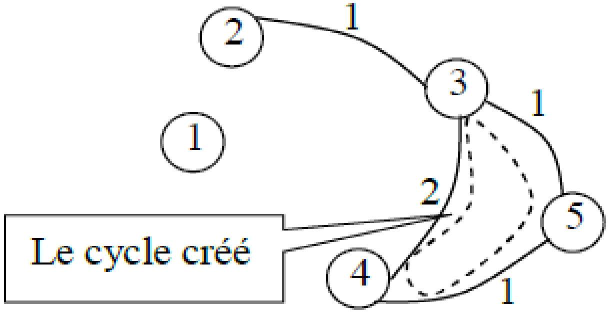
Le graphe (X,W) ci contre ne contient pas de cycle, on pose alors

$W = \{(2,3),(3,5),(4,5)\}$ $|W| = 3$. On a: i≠m alors on pose i=i+1=4.



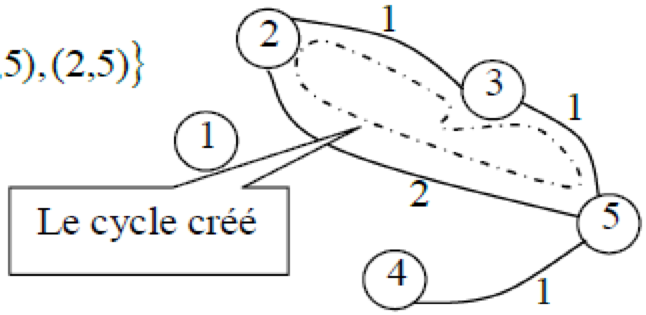
- **Itération 4:**

On a : $u_4=(3,4)$; Soit $W = W \cup \{u_4\} = \{(2,3),(3,5),(4,5),(3,4)\}$
Le graphe (X,W) ci contre contient un cycle,
On a: $i \neq m$ alors on pose $i=i+1=5$.



- **Itération 5:**

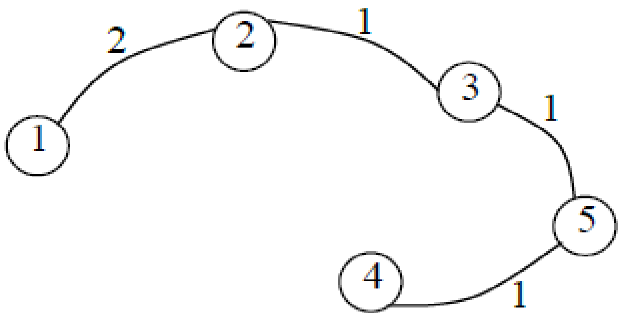
On a : $u_5=(2,5)$; Soit $W = W \cup \{u_5\} = \{(2,3),(3,5),(4,5),(2,5)\}$
Le graphe (X,W) ci contre contient un cycle,
On a: $i \neq m$ alors on pose $i=i+1=6$.



- **Itération 6:**

On a : $u_6=(1,2)$; Soit $W = W \cup \{u_6\} = \{(2,3),(3,5),(4,5),(1,2)\}$
Le graphe (X,W) ci contre ne contient pas de cycle, on pose alors
$W = \{(2,3),(3,5),(4,5),(1,2)\}$ $|W| = n-1 = 5-1 = 4$, terminé

L'arbre du poids minimum est représenté par le graphe A=(X,W).
Le coût total de toute l'installation est: $c(W)=c(u_1)+c(u_2)+c(u_3)+c(u_6)=1+1+1+2=5$.



Le problème de recherche d'un plus court chemin (PCC)

Motivation

Beaucoup de problèmes de la vie quotidienne peuvent être représentés sous forme de graphes…

Le calcul de distance (et donc un plus court chemin) en est un des plus courants :

- **Les logiciels de GPS calculant des itinéraires routiers**
- **Distribution de chaleur dans les alentours**
- **Connexion à haut débit par câble**
- **Routage dans des réseaux de télécommunications**
- **…**

Quelques définitions

Réseau:

Un réseau est un graphe G=(X,U) muni d'une application d :U ->R qui à chaque arc fait correspondre sa longueur L, on note un tel réseau par: R=(X,U,L). En pratique, L(u) peut matérialiser un coût, une distance, une durée, ..etc.

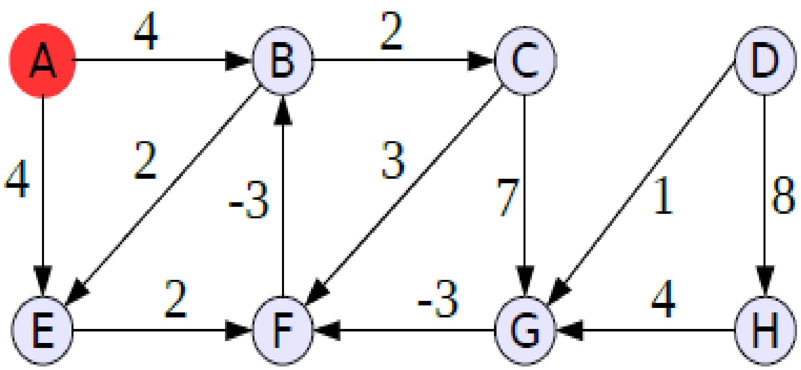
Quelques définitions

Définitions :

- La longueur d'un chemin est la somme des poids des arcs
- La distance entre x et y (*noté, d(x,y)*) est le minimum des longueurs sur tous les chemins.
- Un plus court chemin entre x et y est un chemin dont la longueur est égale à d(x,y).

Exemples :

- Longueur de (A,E,F,B) est $4 + 2 + (-3) = 3$
- $d(A, B) = 3$
- Plus court chemin entre A et B : (A, E, F, B)

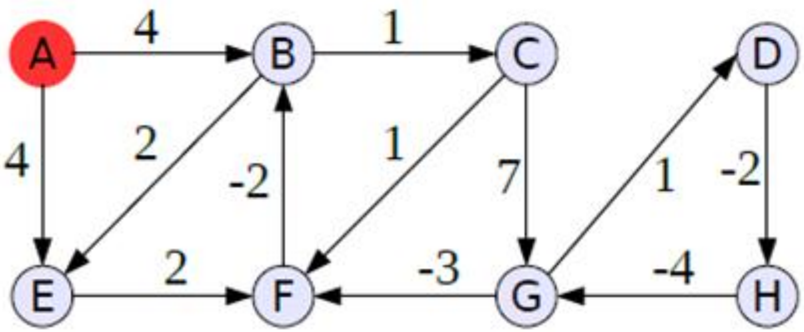


Remarques

Etant donnés deux sommets x et y, plusieurs cas se présentent :
1) il n'y a pas de chemin de x à y.
2) il existe un ou plusieurs plus courts chemins de x à y.
3) il existe des chemins de x à y mais pas de plus court.

Exemples :

1) il y a pas de chemins entre H et A (donc, pas de plus court chemin)
2) il existe deux plus courts chemins entre A et B : (A,B) et (A,E,F,B)
3) il existe une infinité de plus courts chemins entre B et F : (B,C,F), (B,C,F,B,C,F)....



4) Il existe des chemins entre D et G mais pas de plus court : les chemins (D,H,G,D,H,G....) sont arbitrairement courts.

Circuit absorbant

Définition :
Un circuit absorbant est un circuit de longueur négative.

• Si un graphe possède un circuit absorbant, alors il n'existe pas de plus court chemin entre certains de ses sommets.

Théorème : Soit G un graphe orienté pondéré n'ayant pas de circuits absorbants, et x et y deux sommets de G. Si il existe un chemin allant de x à y, alors la distance d(x,y) est bien définie et il existe au moins un plus court chemin d e x à y.

Propriétés des plus courts chemins

Propriété 1 : Tout sous-chemin d'un plus court chemin est un plus court chemin.

Propriété 2 : Si il existe un plus court chemin entre deux sommets x et y, alors il existe un plus court chemin élémentaire (sans cycle) entre x et y.

Algorithmes de résolution de problème de PCC

- Selon les propriétés du graphe traité (orienté/non orienté, avec/sans circuit ou longueurs positives/quelconques) et selon le problème considéré (recherche du plus court chemin d'un sommet vers tous les autres, ou entre tous les couples de sommets), il existe de nombreux algorithmes permettant l'obtention d'une solution.

Algorithmes de résolution de problème de PCC

Algorithmes	Type du PCC	Propriétés du graphe	
		Type de graphe	Longueur
Dijkstra	**D'un sommet à tous les autres sommets**	**Graphe orienté (et non orienté)**	**Longueur positives**
Bellman		**Graphe orienté sans circuit (sommet d'origine doit être sans prédécesseur)**	**Longueur quelconque (nombre réel)**
Bellman-Ford		**Graphe orienté**	
Floyd	**Entre tous les couples de sommets**	**Graphe orienté sans circuit absorbant**	

ALGORITHME DE DIJKSTRA " longueur positive"

- Cet algorithme permet de calculer le PCC d'un sommet « s » à un sommet « d » ou d'un sommet « s » à tous les autres sommets dans un graphe de longueur positive.
  - Soit $\pi(i)$ la valeur de chemin du sommet « s » vers le sommet « i », ainsi, initialement : $\pi(s) = 0$ et $\pi(x) = \infty$ pour tout sommet $x \neq s$
  - Soit M l'ensemble des sommets marqués, initialement il est vide ($M = \phi$)

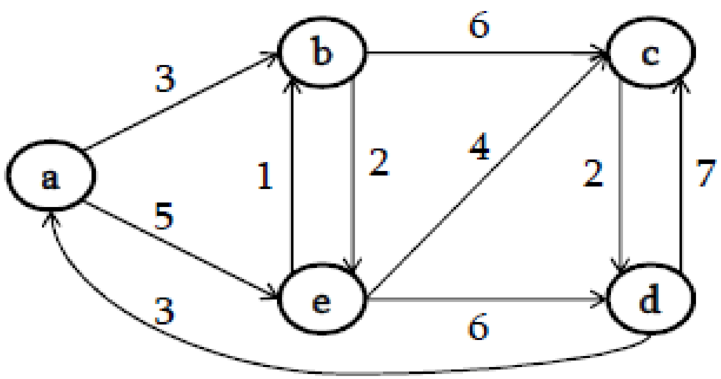
Algorithme de Dijkstra " longueur positive"

- Tant qu'il existe un sommet non marqué ($M \neq X$) ou on n'a pas arrivé au sommet destinataire ($x \neq d$) faire:

1. Choisir un sommet non marqué, soit x ($x \in X\text{-}M$), ayant le plus petit π [$\pi(x) = \min \{\pi(y)$ tq $x \in X\text{-}M\}$]
2. Mettre à jours ses successeurs non encore marqués comme suit: $\pi(y) = \min(\pi(y), \pi(x)+l_{xy})$ tel que $y \in \Gamma^+(x) \cap (X\text{-}M)$
3. Marquer le sommet x [$M = M \cup \{x\}$]

ALGORITHME DE DIJKSTRA " longueur positive"

- **Exemple:** Trouver PCC de a vers tous les autres sommets

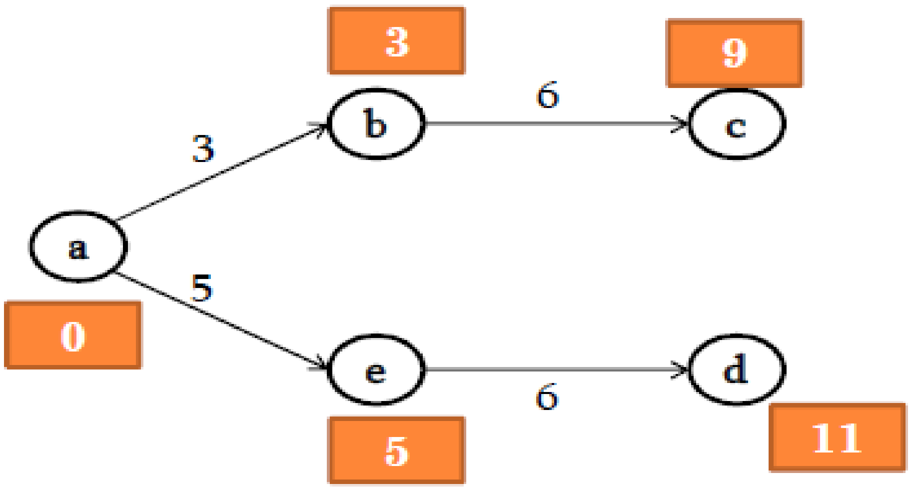


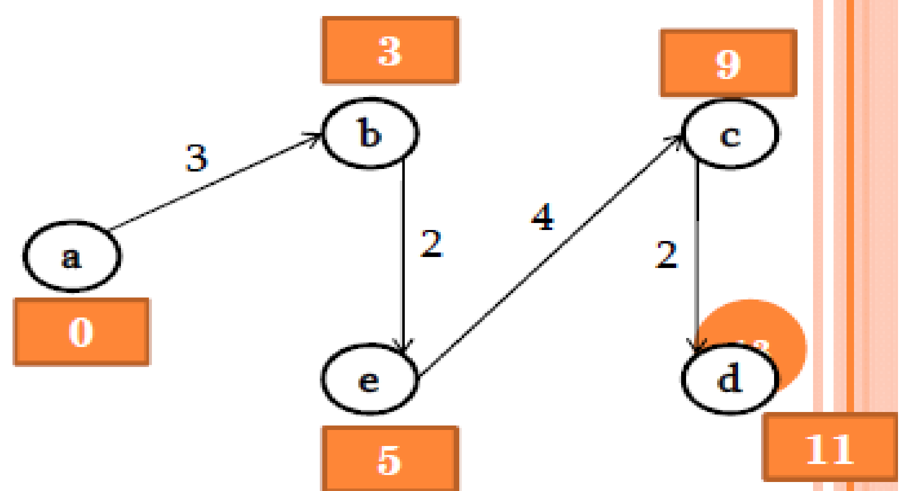
	π(a)	π(b)	π(c)	π(d)	π(e)
0 (init)	0	∞	∞	∞	∞
1	0 (*)	3			5
2		3(*)	9		5
3			9	11	5 (*)
4			9(*)		
5				11(*)	
6 (fin)	0	3	9	11	5

Algorithme de Dijkstra " longueur positive"

Pour chaque couple de sommet (i, j), on garde l'arc vérifiant la relation suivante: u(i,j) = d(j) - d (i).

- **On peut trouver plusieurs arborescences:**





ALGORITHME DE BELLMAN" sans circuit"

- Cet algorithme permet de calculer le PCC d'un sommet « s » à un sommet « d » ou d'un sommet « s » à tous les autres sommets dans un graphe orienté **sans circuit** de longueur quelconque.
  - Soit $\pi(i)$ la valeur de chemin du sommet **sans prédécesseur** « s » vers le sommet « i », ainsi, initialement : $\pi(s) = 0$
  - Soit M l'ensemble des sommets marqués, initialement il contient « s » ($M = \{s\}$)

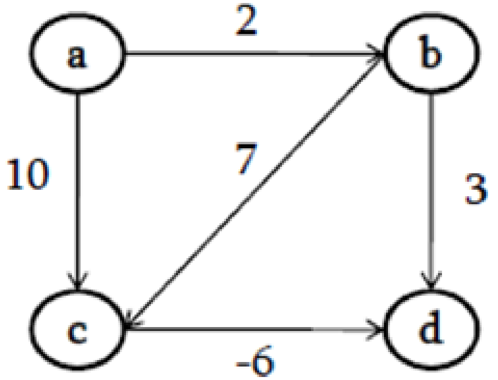
ALGORITHME DE BELLMAN" sans circuit"

- Tant qu'il existe un sommet non marqué ($M \neq X$) ou on n'a pas arrivé au sommet destinataire ($x \neq d$) faire:

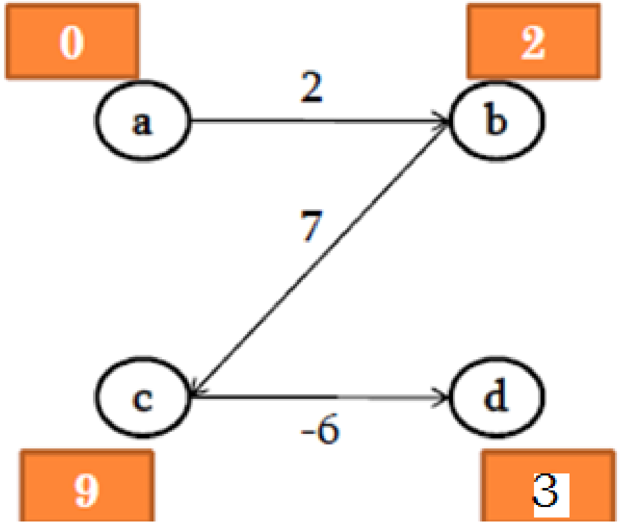
1. Choisir un sommet non marqué, soit x ($x \in X$-M), dont tous les prédécesseurs sont marqués [$\Gamma^{-}(x) \subset M$]
2. Mettre à jours son poids π comme suit: $\pi(x) = \min (\pi(y)+l_{yx})$ tel que $y \in \Gamma^{-}(x)$
3. Marquer le sommet x [$M = M \cup \{x\}$]

ALGORITHME DE BELLMAN" sans circuit"

- **Exemple:** Trouver PCC de a vers tous les autres sommets



	π(a)	π(b)	π(c)	π(d)
0 (init)	0 (*)	-	-	-
1		2 (*)		
2			9(*)	
3				3 (*)
fin	0	2	9	3



Algorithme de Bellman-Ford

- Cet algorithme permet de calculer le PCC d'un sommet « s » à tous les autres sommets dans un graphe orienté de longueur quelconque et aussi de détecter la présence d'un circuit absorbant.

```
fonction Bellman-Ford(G = (S, A), poids, s)
  pour u dans S faire
  |       d[u] = +∞
  |       pred[u] = null
  d[s] = 0
  //Boucle principale
  pour k = 1 jusqu'à taille(S) - 1 faire
   |      pour chaque arc (u, v) du graphe faire
   |      |    si d[u] + poids(u, v) < d[v] alors
   |      |    |    d[v] := d[u] + poids(u, v)
   |      |    |    pred[v]:= u

  retourner d, pred
```

Algorithme de Bellman-Ford

il y a un cycle de poids négatif si et seulement si un nouveau tour de boucle fait diminuer une distance.

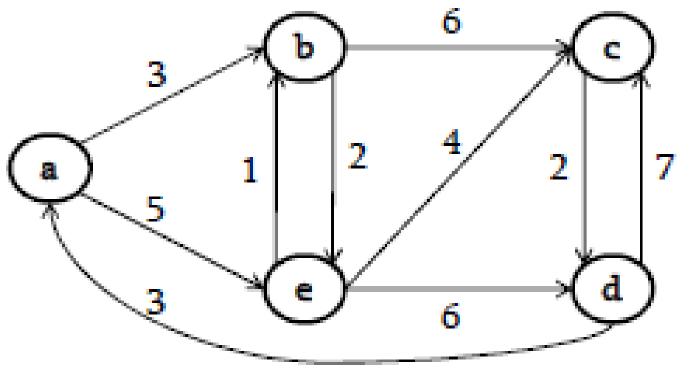
Ainsi, à la fin de l'algorithme, on fait :

```
pour chaque arc (u, v) du graphe faire
 |    si d[v] > d[u] + poids(u, v) alors
 |         afficher "il existe un cycle absorbant"
```

- En absence de circuit absorbant dans le graphe, l'algorithme se termine nécessairement à l'issue de l'itération n (k = n) car, au pire des cas, le PCC de s vers tous les autre sommets est un chemin élémentaire possédant (n-1) arcs.

Algorithme de Bellman-Ford

- **Exemple 1:** Trouver PCC de a vers tous les autres sommets



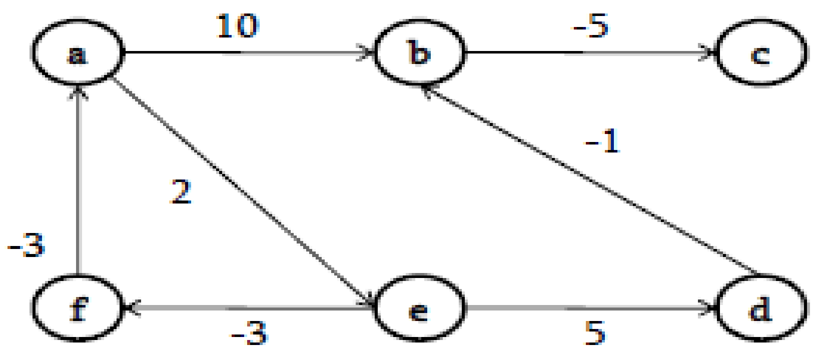
les arc à parcourir chaque iteration : Ab ae bc be eb ed ec cd dc da

	a	b	c	d	e
init	0	∞	∞	∞	∞
1	0	3	9	11	5
2(fin)	0	3	9	11	5
3					
4					
circuit absorbant	Pas de circuit absorbant				

∞

ALGORITHME DE BELLMAN-FORD

- **Exemple 2:** Trouver PCC de a vers tous les autres sommets



les arc à parcourir chaque iteration : ab ae bc ed ef db fa

	a	b	c	d	e	f
init	0	∞	∞	∞	∞	∞
1	0/-4	10/6	5	7	2	-1
2	-8	6	1	3	-2	-5
3	-12	2	-3	-1	-6	-9
4	-16	-2	-7	-5	-10	-13
5	-20	-6	-11	-9	-14	-17
circuit absorbant	Existence d’un circuit absorbant					

Algorithme de Floyd

- Cet algorithme permet de calculer le PCC entre tous les couples de sommets dans un graphe orienté **sans circuit absorbant** de longueur quelconque.
  - Numéroter les sommets de 1 à n ($|X| = n$)
  - Soit la matrice $A = \{a_{ij}\}$ de taille n x n définie initialement comme suit:

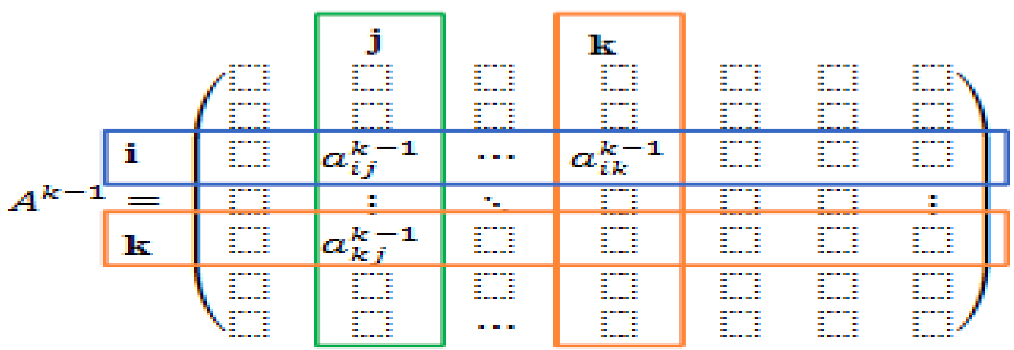
$$a_{ij} = \begin{cases} 0 & \text{si } i = j \\ l_{ij} & \text{si } (i,j) \in U \\ +\infty & \text{sinon} \end{cases}$$

Algorithme de Floyd

fonction *FloydWarshall* (G)
  $\mathcal{W}^0 :=$ matrice d'adjacence de G (matrice $n \times n$)
  for $k := 1$ to n
    for $i := 1$ to n
      for $j := 1$ to n
        $\mathcal{W}_{ij}^{k} = \min(\mathcal{W}_{ij}^{k-1}, \mathcal{W}_{ik}^{k-1} + \mathcal{W}_{kj}^{k-1})$
  renvoyer $\mathcal{W}^n$

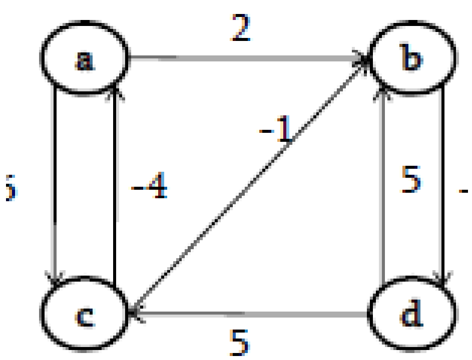
- Voici une description formelle de l'algorithme :
  - Pour tout sommet k (k allant de 1 à n)
    - Pour tout couple de sommet (i, j) calculer

$$a_{ij}^{k} = \min(a_{ij}^{k-1}, \quad a_{ik}^{k-1} + a_{kj}^{k-1})$$



Algorithme de Floyd

- **Exemple:** Trouver PCC entre tous les couples des sommets



$$A^0 = \begin{pmatrix} 0 & 2 & 6 & \infty \\ \infty & 0 & \infty & -2 \\ -4 & -1 & 0 & \infty \\ \infty & 5 & 5 & 0 \end{pmatrix} \qquad A^1 = \begin{pmatrix} 0 & 2 & 6 & \infty \\ \infty & 0 & \infty & -2 \\ -4 & \mathbf{-2} & 0 & \infty \\ \infty & 5 & 5 & 0 \end{pmatrix}$$

$$A^2 = \begin{pmatrix} 0 & 2 & 6 & \mathbf{0} \\ \infty & 0 & \infty & -2 \\ -4 & -2 & 0 & \mathbf{-4} \\ \infty & 5 & 5 & 0 \end{pmatrix} \qquad A^3 = \begin{pmatrix} 0 & 2 & 6 & 0 \\ \infty & 0 & \infty & -2 \\ -4 & -2 & 0 & -4 \\ \mathbf{1} & \mathbf{3} & 5 & 0 \end{pmatrix}$$

$$A^4 = \begin{pmatrix} 0 & 2 & \mathbf{5} & 0 \\ \mathbf{-1} & 0 & \mathbf{3} & -2 \\ -4 & -2 & 0 & -4 \\ 1 & 3 & 5 & 0 \end{pmatrix}$$

Algorithme de Floyd