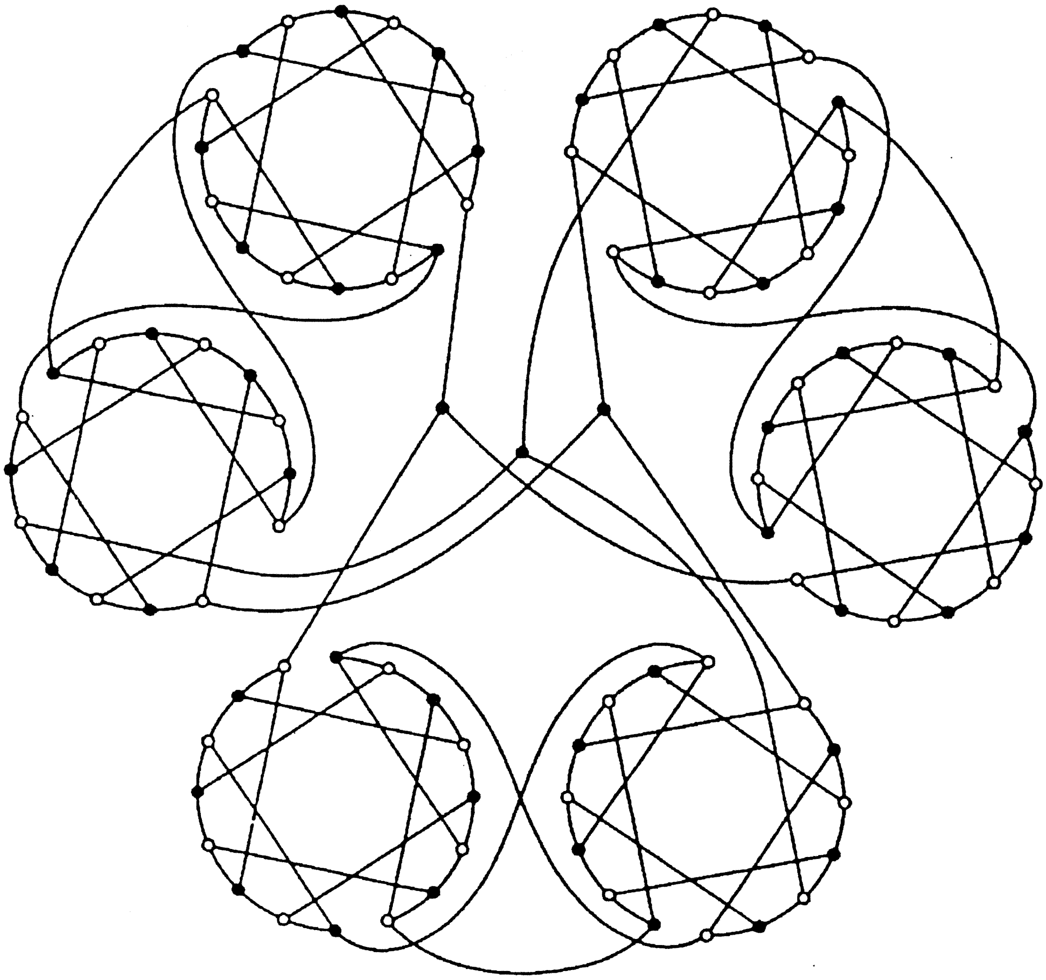
Publication dans le cadre du Thème 3 pour 1997-1998

Publication n°89

LA THEORIE DES GRAPHES AU BTS INFORMATIQUE DE GESTION



Le graphe de Horton.

Direction de l'enseignement scolaire
Bureau A11
Valorisation des innovations pédagogiques

Commission Inter-IREM
Lycées Technologiques
I.R.E.M.
Institut Galilée
av. J.B. Clément
93430 VILLETANEUSE

UNIVERSITE PARIS-NORD - IREM
LA THEORIE DES GRAPHES
AU BTS INFORMATIQUE DE
GESTION - 105 pages

Bernard DORNE

et Jean-Marie VANHERPE

ISBN 286240 089 0

Dépôt légal : 2ème trimestre 1998

300 ex
50.00F.

Cette brochure a été réalisée dans le cadre de travaux de recherche lancés conjointement par l'Adirem et la Direction de l'enseignement scolaire.

Il s'agit d'un document d'information sur la théorie des graphes destiné aux professeurs de mathématiques enseignant en Bts Informatique de gestion pour accompagner le nouveau programme de mathématiques dans ce Bts applicable dès la session 1998.

B. VERLANT
Responsable de la commission
inter-IREM «lycées technologiques»
et du «thème 3» de la convention
Adirem - DLC

Table des matières.

Avant propos.

Ce document est la conséquence de l'apparition des premiers éléments de théorie des graphes au BTS d'Informatique de gestion au sein du nouveau programme Mathématiques.
Il est organisé comme suit :

Une présentation théorique

Elle se veut suffisamment simple pour être accessible au non-spécialiste de la théorie des graphes et suffisamment générale pour lui permettre d'avoir une vision globale du sujet. De ce fait elle déborde largement du cadre du programme et s'adresse donc aux enseignants plutôt qu'aux étudiants.
Elle comporte :
Une rapide introduction, qui présente la théorie des graphes et fait état d'une notion importante quoique hors programme que l'enseignant doit avoir présente à l'esprit: la notion de complexité d'un algorithme.
Suit une partie consacrée au vocabulaire et définitions courantes.
Des algorithmes pour la plupart matriciels traitent ensuite des problèmes des chemins de longueur fixée, de la fermeture transitive, du problème du plus court chemin et du calcul du rang d'un sommet.

Des applications pédagogiques

Un choix d'exercices que l'on s'est efforcé d'adapter au niveau des étudiants de BTS avec le souci :
D'associer étroitement graphes et calcul matriciel.
De montrer que les problèmes relatifs aux graphes peuvent se traiter à l'aide de l'outil informatique.
De proposer des exemples d'utilisation des graphes en gestion (problèmes de choix multi-critères ; méthode Electre, problèmes d'ordonnancement des tâches : méthode PERT).
Les exercices et thèmes d'étude sont donc de type et de niveaux variés :
Des exemples concrets, des méthodes pratiques permettant à l'étudiant de s'approprier les notions de base.
Des algorithmes dont les déroulements ont été explicités sur des exemples pour en faciliter la compréhension
Des applications de gestion.

Une annexe
A :l'extrait du programme de BTS relatif aux graphes.
B :Une bibliographie.

Concluons en citant «Toute pensée originale procède par images[1]» Aussi gageons que l'étude des graphes dont la compréhension naît de l'image sera une occasion pour l'étudiant de BTS de prendre goût aux activités où s'imbriquent mathématiques, informatique et gestion.

.

[1] Les graphes par l'exemple, F.Droesbeke, Avant propos.

Première partie .

- **Une présentation théorique.**

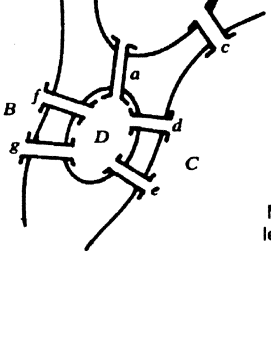
I. Introduction.

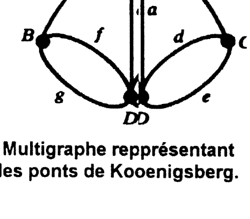
Un graphe peut être vu comme la donnée d'une collection d'objets, les sommets, et de liens entre ces sommets: les arcs ou les arêtes. L'adage «un bon dessin vaut mieux qu'un long discours», renforce l'idée que de la théorie des graphes est un outil simple de modélisation pour de nombreux problèmes.
Primitivement introduite en 1736 par Euler pour résoudre le problème des ponts de Koenigsberg,[2] la théorie des graphes est restée longtemps cantonnée à la résolution de curiosités mathématiques.
L'arrivée des calculateurs électroniques et plus généralement de l'informatique a fait naître de nouveaux problèmes qui s'expriment naturellement en termes de graphes. C'est le cas par exemple des problèmes d'optimisation du code des compilateurs qui se ramènent à un problème de coloriage optimal d'un graphe; prévenir les «dead-locks» dans un système d'exploitation revient à détecter des circuits dans un graphe de dépendance; ou encore l'étude de la fiabilité des réseaux commence par celle de la connexité des graphes...

Dès que l'on cherche à automatiser la résolution de tels problèmes, dès que l'on écrit des algorithmes destinés à être implantés sur des machines, la question de l'efficacité de ces algorithmes se pose. Il se trouve en effet que certains problèmes (pas seulement en théorie des graphes) sont sans solution algorithmique ou ont des solutions tellement «complexes» qu'ils sont irréalisables en pratique par une machine dès que leur «taille» devient importante.
Un exemple classique est le problème de Satisfiabilité: *Existe-t-il une façon d'attribuer une valeur aux variables d'une expression booléenne donnée sous forme factorisée de telle sorte que cette expression booléenne soit vraie?*
Une solution évidente vient immédiatement à l'esprit, il suffit de donner une valeur booléenne à chaque variable et de vérifier en calculant l'expression, en faire la table de vérité en quelque sorte; cependant on assiste à une explosion combinatoire lorsque que la taille devient importante (le nombre de cas à étudier est de l'ordre de 2^n où n est le nombre de variables) qui fait que ce problème est réputé «intraitable» et le reste encore actuellement.

---

[2] **Le problème des ponts de Koenigsberg**: Existe-t-il un parcours de la ville empruntant une fois et une seule chacun des 7 ponts de la ville?
Euler a démontré qu'un multigraphe admet un parcours empruntant exactement chaque arête une seule fois si et seulement si chaque sommet est de degré pair à l'exception peut -être de 2 sommets, ce qui montrait que le problème des ponts de Koenigsberg n'avait pas de solution.





Multigraphe repprésentant les ponts de Kooenigsberg.

On peut considérer que ce problème connu sous le nom de problème SAT est à l'origine de la théorie de la complexité, la notion de complexité est suffisamment importante pour que l'on s'y attarde.[3].

Complexité d'un algorithme.

Dès que l'on écrit un algorithme destiné à être implanté sur une machine, on cherche à mesurer son efficacité. Cette «mesure» se doit d'être indépendante de la machine ou du langage de programmation utilisés mais doit au contraire tenir compte de la «taille» du problème traité.

Les critères les plus répandus servant à caractériser un algorithme sont le temps maximum nécessaire à son exécution mesuré en nombre d'opérations élémentaires ou l'espace maximum nécessaire mesuré en unités de stockage élémentaires, on parle alors de complexité en temps ou de complexité en espace, dans le pire des cas.

La mémoire des ordinateurs étant de moins en moins coûteuse c'est la complexité en temps qui devient importante actuellement et c'est du temps d'exécution que l'on cherche à gagner lorsque l'on optimise un algorithme.
Arguant du fait que le pire des cas n'est pas toujours le plus intéressant, certaines analyses fournissent une complexité en temps moyen d'exécution s'appuyant sur des arguments probabilistes.

Quoi qu'il en soit la complexité d'un algorithme est une fonction n->f(n) de la taille n de la donnée, et c'est souvent au comportement asymptotique de cette fonction, à son ordre de grandeur que l'on s'intéresse. C'est pourquoi la complexité d'un algorithme sera donnée sous la forme O(g(n)) où g est une fonction de n telle que

$$\exists\lambda \in R; \exists n_0 \in N; \forall\ n \geq n_0\ f(n) \leq \lambda\ g(n)$$

A titre d'exemple les deux algorithmes suivants ont respectivement une complexité en temps dans le pire des cas de $O(n^2)$ et O(n) lorsque n représente la taille de la donnée. Le premier initialise une matrice carrée d'ordre n tandis que le second calcule le plus grand terme d'une liste de n nombres entiers naturels. A titre de comparaison l'algorithme trivial pour le problème SAT donné précédemment a une complexité en temps dans le pire des cas de $O(2^n)$.

[3]S.A Cook, *The Complexity of theorem-proving procedures*, Proc. 3rd Annual ACM Symposium on Theory of Computing, pp 151-158,(1971).
M.R.Garey et D.S.Johnson, *Computers and Intractability: a guide to the Theory of NP-Completeness*, Freeman,San Francisco, CA, (1979)

Algorithme *Initialise une matrice.*

Donnée: Une matrice carrée A d'ordre n

Résultat. La matrice A est initialisée (chaque terme est affecté d'une valeur)

Pour chaque ligne i de la matrice
    Pour chaque colonne j de la matrice
        initialiser le terme A[i, j].(Celui de la $i^{ème}$ ligne et de la $j^{ème}$ colonne).

Algorithme *Calcule le maximum.*

Donnée: Une liste non vide L de n nombres entiers naturels.

Résultat: Le plus grand élément de la liste L.

Initialiser la variable Max à 0.
Pour chaque terme x de la liste L
    Si x>Max alors affecter à Max la valeur de x.
Retourner la valeur de Max.

On peut noter qu'il n'y a pas de forme homologuée pour décrire un algorithme, on peut utiliser une description purement littéraire comme cela a été fait pour le problème SAT ou bien adopter une forme plus structurée comme ci-dessus. Il faut savoir que trop codifier la description reviendrait à utiliser un langage de programmation mais qu'il est des cas où l'on ne peut échapper à une description précise de la structure de données dont dépend souvent la complexité.

En matière de graphes la taille de la donnée dépend de deux paramètres qui sont le nombre de sommets désigné par n dans la suite et le nombre d'arcs (ou d'arêtes) (m dans la suite).

En algorithmique séquentielle, on considère comme optimal un algorithme sur les graphes en O(n+m).

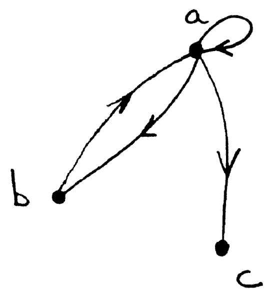
II. Vocabulaire et définitions.

Il est très difficile de donner une définition unique pour un graphe en ce sens que sous le concept unique de la donnée de sommets et de liens entre ces sommets peuvent se présenter différents types de graphes pas toujours compatibles. Un graphe peut être orienté (les liens induisent un ordre sur les sommets) ou non, il peut être simple (un seul lien possible entre des sommets) ou multiple, il peut être avec ou sans boucle (aucun sommet n'est relié à lui-même), il peut être valué (une valeur est associée à chaque lien) ou/et pondéré (une valeur est associée à chaque sommet), on peut aussi parler d'hypergraphe (les liens sont des ensembles de sommets) et même de graphe infini. Aussi nous ne donnerons pas une définition de graphe mais nous dresserons plutôt une liste des définitions les plus courantes dans la littérature sachant que le concept qui prévaut au BTS d'informatique de gestion est celui de graphe simple orienté.

Définition: *Graphe simple orienté.*[4]
Un graphe simple orienté G=(X,U) est déterminé par la donnée de:

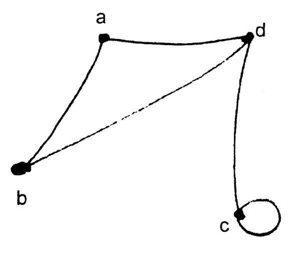
un ensemble X dont les éléments sont appelés les sommets du graphe.

un ensemble U, de couples de X^2 dont les éléments sont appelés arcs.



Dans l'exemple, les sommets de G sont a, b et c.
Les arcs sont (b,a),(a,b),(a,a) et (a,c).

On peut aussi définir un graphe simple orienté par la donnée du couple (X,Γ) où X est un ensemble de sommets et Γ une application qui à tout sommet x de X associe une partie de X : l'ensemble des successeurs de x.



Définition: *Graphe simple non orienté*[5]
Un graphe simple non orienté G=(X,U) est déterminé par :

Un ensemble fini et non vide X dont les éléments sont les sommets du graphe.

Un ensemble U de paires de sommets dont les éléments sont appelés arêtes.

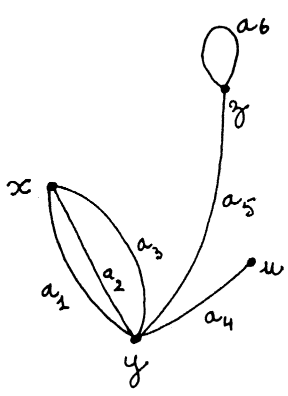
Dans l'exemple, les sommets de G' sont a, b, c, d et les arêtes sont ab, ad, cc, db, cd

Certains auteurs voient un graphe comme une relation binaire (concept orienté) et définissent un graphe non orienté comme une relation binaire symétrique alors que pour d'autres un graphe est par essence non orienté et un graphe orienté n'est rien d'autre qu'un graphe dont les arêtes sont munies d'une orientation.

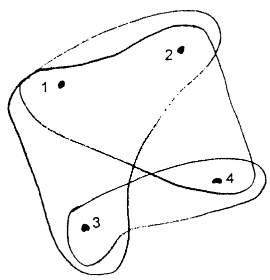
---

[4] M. Gondran et M Minoux, *Graphes et algorithmes*, Ed Eyrolles 1994.
[5] A.V.Aho,J.E.Hopcroft,J.D.Ullman, *The Design and Analysis of Computer Algorithms,* Addison-Wesley (1974)

Définition: *Multigraphe.*[6]
Un Multigraphe G=(V,E,ψ) est déterminé par:
un ensemble non vide V dont les éléments sont les sommets du graphe.
un ensemble E dont les éléments sont des arêtes.
une application ψ qui à chaque arête de E associe une paire non ordonnée de sommets pas forcément distincts.



Dans l'exemple, les sommets du multigraphe sont x,y,z, et u. Ses arêtes sont a_1, a_2, a_3, a_4, a_5 et a_6, tandis que $\psi(a_1)=\psi(a_2)=\psi(a_3)=xy$, $\psi(a_4)=uy$, $\psi(a_5)=yz$ et $\psi(a_6)=zz$.



Définition: *Hypergraphe*[7]
Un hypergraphe H= (X,U) est déterminé par:
un ensemble X dont les éléments sont les sommets de l'hypergraphe.
un ensemble U de parties de X dont les éléments sont les hyperarêtes.

Dans l'exemple les sommets de l'hypergraphe sont 1,2,3 et 4 et les hyperarêtes sont les ensembles {3,4}, {1,2,3} et {1,2,4}.

Evidemment de nombreuses variantes sont possibles, il en est de même pour le vocabulaire qui suit : les définitions peuvent différer d'un auteur à l'autre.

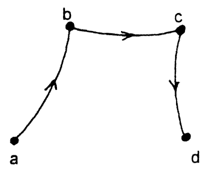
Vocabulaire.

Etant donné un graphe simple fini orienté G=(X,U), 2 sommets sont dits **adjacents** s'il existe un arc les reliant (une **arête** pour un graphe non orienté); dans l'arc (a,b) le sommet a est l'**origine** et le sommet b est l'**extrémité** (dans le cas non orienté on parle uniquement d'**extrémités**); on dit que b est un **successeur** de a et que a est un **prédécesseur** de b (prédécesseur et successeur sont les **voisins** d'un sommet). On a une **boucle** si l'origine et l'extrémité sont confondus.
L'ensemble des successeurs d'un sommet x est désigné par $\Gamma^+(x)$ (ou $\Gamma(x)$), $\Gamma^+(x)=\{y\,/\,y\in X;(x,y)\in U\}$, l'ensemble des prédécesseurs est désigné par $\Gamma^-(x)$ (ou $\Gamma^{-1}(x)$), $\Gamma^-(x)=\{y\,/\,y\in X;(y,x)\in U\}$. Par abus de langage, $\Gamma^+(x)$ et $\Gamma^-(x)$ sont appelées respectivement «fonction» successeur et «fonction» prédécesseur (dans le cas non orienté l'ensemble des voisins d'un sommet est le voisinage, noté V(x), $V(x)=\{y\,/\,y\in X;xy\in U\}$, l'ensemble des **non-voisins** est le **non-voisinage** $\overline{V(x)};\overline{V(x)}=X-V(x)$). Le **degré sortant** (ou demi degré extérieur) (noté $d^+(x)$) d'un sommet x est le nombre de ses successeurs et le **degré entrant** (ou demi degré

[6] J.A.Bondy and U.S.R.Murty *Graph Theory with Applications*, North-Holland, New York, (1979)
[7] C.Berge, *Graphes et hypergraphes*, Ed Dunod,(1973)

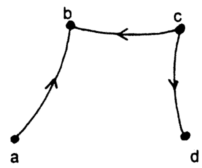
intérieur)(noté $d^-(x)$) est le nombre de ses prédécesseurs (on parle de **degré** pour dénombrer les voisins).

Une suite finie de sommets $(x_i)_{i=0..k}$ pour $k \geq 1$ telle que $x_{i-1}x_i$ est un arc pour tout i=1..k est un **chemin** de longueur k .



Un chemin (a,b;c;d)

Un chemin dont tous les sommets (à l'exception peut-être du premier et du dernier) sont distincts est **élémentaire**. Un chemin dont le premier et le dernier sommet sont confondus est un **circuit**



Une chaîne: ((a,b),(c,b),(c,d))

Une **chaîne** est une séquence d'arcs tous différents telle que 2 arcs consécutifs ont une extrémité commune.

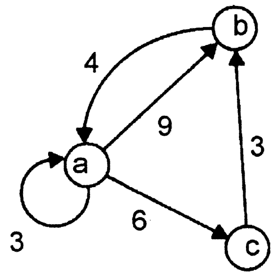
Un **cycle** est une chaîne dont le premier et le dernier arc ont une extrémité commune. Dans le cas non orienté les concepts de chemin et de chaîne coïncident de même que circuit et cycle.

S'il existe une application v de l'ensemble des arcs (ou arêtes) vers l'ensemble des nombres réels, le graphe est dit **valué** par v on parle alors de la **valeur** (ou longueur) d'un arc (ou d'une arête). Le concept de longueur d'un chemin ou d'un circuit peut être adapté à la valuation, et la longueur d'un chemin est alors la somme des valeurs des arcs qui le constituent.

Un chemin ou un circuit est **hamiltonien** s'il emprunte une fois et une seule chacun des sommets du graphe. Il est à noter que si le problème du parcours eulérien peut -être considéré comme résolu (existe-t-il un parcours du graphe empruntant chaque arc une fois et une seule?) celui du parcours hamiltonien (existe-t-il un chemin hamiltonien dans un graphe?) est ouvert, c'est même un problème réputé intraitable au même titre que le problème SAT.

Exemple:

La figure ci-contre représente un graphe simple orienté valué.
Les successeurs de a sont: a,b,c.
Les prédécesseurs de a sont: : a et b.



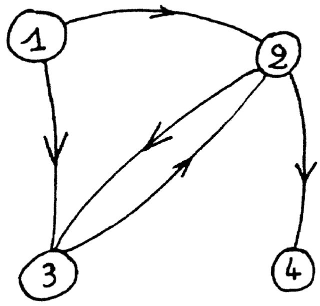
(a,a) est une boucle de valeur 3.
(c,b,a,b) est un chemin non élémentaire de longueur 16 (au sens de la valuation).
(a,c,b,a) est un circuit élémentaire de longueur 13 (au sens de la valuation).
(c,b,a,c) est un circuit hamiltonien.
((c,b),(a,b),(b,a)) est une chaîne de longueur 3 (au sens du nombre d'arcs).
((a,c),(b,a),(c,b)) est un cycle.

III. Représentations d'un graphe en machine.

La représentation picturale d'un graphe que peut faire un être humain ne peut être utilisée telle quelle par une machine, des structures de données plus adaptées sont nécessaires. Selon l'implantation choisie, les opérations élémentaires comme déterminer si deux sommets sont ou non adjacents, calculer le degré d'un sommet, supprimer un arc auront des complexités différentes.

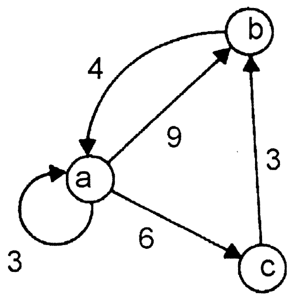
Tableau des successeurs (ou des prédécesseurs).

On peut représenter un graphe sous la forme d'un tableau de deux colonnes contenant pour l'une les sommets du graphe et pour l'autre **la liste** de ses successeurs



Sommets	Successeurs
1	3,2
2	3,4
3	2
4	

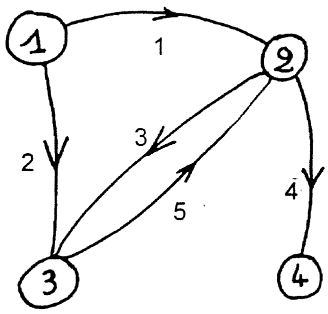
Cette implantation fondée sur la gestion de listes offre un aspect dynamique bien adapté aux opérations élémentaires sur les graphes. Autrement connue sous le nom d'implantation par listes d'adjacence, elle est considérée comme optimale sur le plan de la complexité en espace.
Cette représentation est utilisable avec des multigraphes comme avec des graphes simples, des graphes non orientés (considérés comme des graphes orientés symétriques) et des graphes avec boucles. On peut choisir de fournir la liste des prédécesseurs au lieu des successeurs et d'y inclure une éventuelle valuation.
L'inconvénient de cette représentation est que l'accès direct n'est offert que pour les sommets, on se doit de parcourir de façon séquentielle la liste des successeurs (ou des prédécesseurs) d'un sommet pour accéder aux arcs.



Sommets	Prédécesseurs
a	(a,3), (b,4)
b	(a,9),(c,3)
c	(a,6)

Matrice d'incidence sommets-arcs.

C'est une matrice $(a_{ij})_{i=1..n,j=1..m}$ de taille nxm où n est le nombre de sommets et m le nombre d'arcs. Chacune des n lignes représente un sommet et chacune des m colonnes représente un arc. Le coefficient a_{ij} peut valoir -1, +1 ou 0 selon que le sommet i est l'origine, l'extrémité ou n'a rien à voir avec l'arc j.



$$\begin{pmatrix} -1 & -1 & 0 & 0 & 0 \\ +1 & 0 & -1 & -1 & +1 \\ 0 & +1 & +1 & 0 & -1 \\ 0 & 0 & 0 & +1 & 0 \end{pmatrix}$$

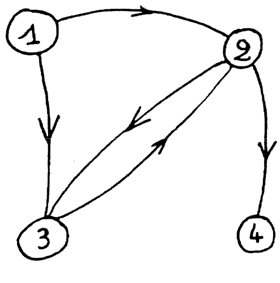
Dans le cas d'un graphe non orienté, la matrice est booléenne.

Cette représentation n'est pas adaptée au cas des graphes avec boucles ni à celui des graphes valués, de plus la matrice est creuse (seuls 2 coefficients par colonne sont non nuls).

Matrice d'adjacence.

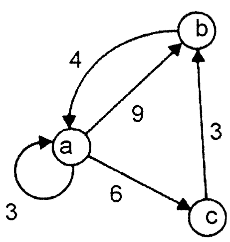
La matrice d'adjacence ou matrice d'incidence sommets-sommets est une matrice booléenne, carrée, d'ordre n $(a_{ij})_{i=1..n,j=1..n}$ (où n est le nombre de sommets). les sommets sont supposés numérotés de 1 à n. Elle est définie par:

Pour i,j=1..n; a_{ij}=1 si et seulement si il existe un arc reliant le sommet i au sommet j.



$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Cette représentation n'est pas utilisable dans le cas des multigraphes valués , pour un graphe non orienté la matrice est symétrique, on peut l'adapter au cas des graphes valués, c'est alors une matrice non booléenne à coefficients réels.



$$\begin{array}{c|c} & \begin{matrix} a & b & c \end{matrix} \\ \begin{matrix} a \\ b \\ c \end{matrix} & \begin{pmatrix} 3 & 9 & 6 \\ 4 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix} \end{array}$$

Les opérations élémentaires sur les graphes et leur coût selon l'implantation

Opération	Tableau des successeurs (listes d'adjacence)	Matrice sommets-arcs	Matrice d'adjacence
Complexité en espace	O(n+m)	O(nxm)	O(n^2)
test de l'existence de l'arc (i,j)	O(degré de i)	O(m)	O(1)
Calcul du degré d'un sommet i	O(degré de i)	O(m)	O(n)
Suppression d'un arc (i,j)	O(degré de i). O(degré de i+ degré de j) en non-orienté	O(m)	O(1)

Coût du passage d'une implantation à une autre.

	Liste d'adjacence	Matrice sommet-sommet	Matrice sommet-arc
Liste d'adjacence		O(n^2)	O(nxm)
Matrice sommet-sommet	O(n^2)		O(nxm)
Matrice sommet-arc	O(nxm)	O(nxm+n^2)	

Un autre exemple d'implantation aussi condensée que les listes d'adjacence utilisant seulement 2 tableaux est montrée en exercice (Thème d'Etude I.8)

.

IV. Problèmes de cheminement.

- Chemins de longueur p.
- Fermeture transitive.
- Le problème du plus court chemin.

Chemins de longueur p.

***Le problème**: Etant donné un graphe simple orienté G et un nombre entier naturel non nul p. Etant donnés 2 sommets i et j du graphe G, existe-t-il un chemin de i vers j de longueur p (au sens du nombre d'arcs)?*

Propriété: Soit G un graphe simple orienté donné par sa matrice d'adjacence A et soit p un nombre entier naturel.
On désigne par $a^{(p)}_{ij}$ le coefficient de la ième ligne et de la jème colonne de la matrice booléenne A^p.(la puissance p ème de A quand la multiplication matricielle est celle induite par le «ou» et le «et» booléens sur les coefficients)

On a alors *$a^{(p)}_{ij}=1$ si et seulement si il existe un chemin de longueur p du sommet i vers le sommet j.*

Preuve: par récurrence sur p.
cas p=1: $A^1=A$ c'est à dire que $a^{(1)}_{ij}=1$ si et seulement si l'arc (i,j) existe
si et seulement si il existe un chemin de longueur 1 du sommet i vers le sommet j.

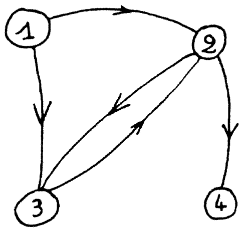
$$\begin{array}{ccc} & k \;\; .. \;\; .. \;\; .. & \begin{array}{c} j \\ \begin{pmatrix} & & & \vdots \\ .. & .. & .. & 1 \\ & & & \vdots \\ & & & \vdots \\ & & & \vdots \end{pmatrix} = A \end{array} \\ i & \underbrace{\begin{pmatrix} & \vdots & & & \\ & \vdots & & & \\ .. & 1 & .. & .. & .. \\ & & & & \end{pmatrix}}_{A^p} & \begin{pmatrix} & & & \vdots \\ & & & \vdots \\ .. & .. & .. & 1 \\ & & & \end{pmatrix} = A^{p+1} \end{array}$$

Considérons que la propriété est vérifiée au rang p et calculons A^{p+1}
Une condition nécessaire et suffisante pour que le coefficient booléen $a^{(p+1)}_{ij}$ soit égal à 1 est qu'il existe un indice k vérifiant $a^{(p)}_{ik}=1$ et $a^{(1)}_{kj}=1$.

Ce qui signifie exactement qu'il existe un sommet k vérifiant:
    Il existe un chemin de i à k de longueur p (hypothèse de récurrence)
    et il existe un arc de k vers j (définition de la matrice d'adjacence)
En d'autres termes $a^{(p+1)}_{ij}=1$ ssi il existe un chemin de longueur p+1 de i vers j.

En conclusion tester de l'existence d'un chemin de longueur p dans un graphe simple orienté revient à calculer la matrice booléenne A^p. Complexité $O(n^3)$ pour p fixé.

Exemple:



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

On peut remarquer qu'aucun des chemins donnés par la matrice A^4 n'est élémentaire.
Plus généralement, les chemins donnés par la matrice A^p ne sont sûrement pas élémentaires, il est donc exclu de vouloir résoudre le problème de l'hamiltonisme en calculant A^{n-1}.

Une approche analogue où l'on ne se contente pas de rechercher l'existence d'un chemin de longueur p mais où l'on dénombre ces chemins est présentée dans la deuxième partie (Thème d'Etude II.1), la matrice que l'on utilise alors n'est plus booléenne.

Fermeture transitive.

Définition*: Soit G un graphe simple orienté ayant n sommets. La fermeture transitive de G est le graphe G' ayant les mêmes sommets que G tel que (x,y) est un arc de G' si et seulement si il existe un chemin du sommet x vers le sommet y dans G.*
C'est en fait le graphe de la relation de descendance induite par les adjacences de G.
Le problème : *Il s'agit de trouver la fermeture transitive d'un graphe simple non orienté.*
Pour traiter ce problème nous considérons qu'un graphe simple orienté G est donné sous la forme G=(X,Γ) où Γ est la «fontion» successeur.Etant donnés 2 graphes G_1=(X, Γ_1) et G_2=(X, Γ_2), on définit les opérations suivantes dans l'ensemble des applications de X vers l'ensemble des parties de X :

$$\Gamma_1 \cup \Gamma_2 \text{ est défirie par: } \forall x \in X;(\Gamma_1 \cup \Gamma_2)(x) = \Gamma_1(x) \cup \Gamma_2(x)$$

$$\Gamma_1 \bullet \Gamma_2 \text{ est défirie par: } \forall x \in X;(\Gamma_1 \bullet \Gamma_2)(x) \text{ est une partie de X telle que:}$$

$y \in (\Gamma_1 \bullet \Gamma_2)(x)$ signifie : *y est **le** successeur dans G_1 **d'un** successeur de x dans G_2*

Propriété : *Si M_1 est la matrice d'adjacence de G_1=(X,Γ_1) et si M_2 est la matrice d'adjacence de G_2=(X,Γ_2)*
Alors M_1+M_2 est la matrice d'adjacence de (X, $\Gamma_1 \cup \Gamma_2$)
et $M_1 \bullet M_2$ *est la matrice d'adjacence de (X, $\Gamma_2 \bullet \Gamma_1$)*

Dans le cas particulier où G_1=G_2=G (Γ_1=Γ_2=Γ) , lorsque M est la matrice booléenne d'adjacence de G , la matrice M^i est la matrice d'adjacence du graphe (X,Γ^i) avec
$y \in \Gamma^i(x)$ si et seulement si il existe dans G un chemin de longueur i de x vers y.

La fermeture transitive de G est donc définie par G'=(X,Γ') avec :

$$\forall x \in X; \Gamma'(x) = \Gamma(x) \cup \Gamma^2(x) \cup \ldots . \cup \Gamma^{n-1}(x)$$

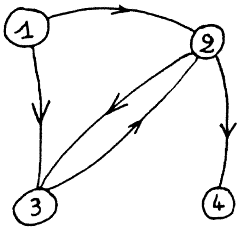
L'existence d'un chemin quelconque entre 2 sommets étant équivalente à l'existence d'un chemin élémentaire entre ces 2 sommets, il est inutile de faire intervenir Γ^k pour k> n-1puisque cela concerne des chemins de longueur supérieure à n-1 qui sont non élémentaires.

La matrice d'adjacence M' de la fermeture transitive de G est donc fournie par:
$M' = M + M^2 + \ldots . + M^{n-1}$

Complexité du calcul $O(n^4)$.

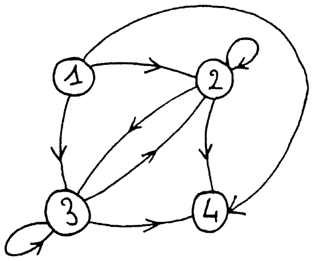
D'autre part :
comme $I + M + M^2 + \ldots + M^{n-2} = (M + I)^{n-2}$
on obtient $M' = M + M^2 + \ldots M^{n-1} = M(M + I)^{n-2}$

Exemple:
Un graphe simple orienté G.



$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} ; A^2 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} ; A^3 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

La fermeture transitive de G



$$A' = A + A^2 + A^3 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Une autre solution matricielle un peu plus efficace est donnée dans la deuxième partie (Algorithme de Warshall).

Ce qui précède permet aussi de résoudre le problème des chemins de longueur p.
On trouvera une utilisation de la fermeture transitive dans l'exercice III.1

Le problème du plus court chemin.

***Le problème**: Etant donné un graphe simple orienté valué. Quel est le plus court chemin permettant d'aller du sommet i au sommet j? (ici la notion de longueur d'un chemin est à comprendre comme la somme des valuations des arcs de ce chemin).*

Une condition nécessaire d'existence du plus court chemin de i à j est l'absence de **circuit** de longueur négative sur tout chemin de i à j.
Par convention, si j n'est pas accessible à partir de i, le plus court chemin de i à j est infini ; d'autre part tout sommet i est accessible à partir de lui-même par un chemin de longueur 0.
Le cas où le graphe est non valué est une instance particulière de celui-ci: il suffit de considérer que tous les arcs sont valués par 1.
Il existe de nombreux algorithmes pour résoudre le problème du plus court chemin, selon que la valuation des arcs est positive, selon que le graphe est sans circuit de longueur négative, selon que l'on recherche les plus courts chemins à partir d'un sommet donné ou pour tout couple de sommets.
Nous présentons ici l'algorithme de Floyd, d'autres algorithmes sont proposés dans la deuxième partie.

Algorithme de Floyd (1962).
On part de la matrice d'adjacence modifiée de la manière suivante:

$A = \left(a_{ij}\right)_{i,j=1..n}$ où les coefficients a_{ij} sont dé finis par:

$$\begin{cases} a_{ij} = \text{ la valeur de l'arc (i,j) s'il existe.} \\ a_{ij} = \infty \text{ si l'arc (i,j) n'existe pas.} \\ a_{ii} = 0. \end{cases}$$

On calcule de façon itérative la matrice $A^{(k)} = (a^{(k)}_{ij})_{i,j=1..n}$ où le coefficient $a^{(k)}_{ij}$ représente la valeur du plus court chemin de i à j **n'utilisant que les k premiers sommets**. Ainsi par exemple $a^{(1)}_{46}$ représente le plus court chemin de 4 à 6 n'utilisant éventuellement que le sommet 1.
Trivialement on a $A^{(0)} = A$ et la solution est fournie par $A^{(n)}$.

Les coefficients de $A^{(k)}$ se calculent à partir de ceux de $A^{(k-1)}$ avec la formule suivante:

$$a^{(k)}_{ij} = \min(a^{(k-1)}_{ij} , a^{(k-1)}_{ik} + a^{(k-1)}_{kj}).$$

Cette formule peut se comprendre de la façon suivante:

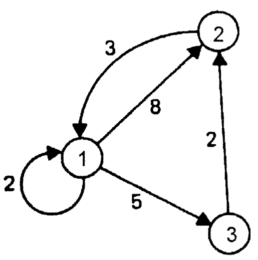
Le plus court chemin pour aller de i à j en n'utilisant que les sommets de 1 à k est le plus petit entre:
Le plus court chemin pour aller de i à j en n'utilisant que les sommets de 1 à k-1 et entre
Le plus court chemin pour aller de i à j en passant par k et en n'utilisant que les sommets de 1 à k-1 pour aller de i à k puis de k à j.

Comme on n'a pas besoin de conserver les différentes matrices obtenues au cours des différentes itérations on en vient à l'algorithme suivant.

Algorithme Plus court chemin (Floyd)
Donnée: Un graphe simple orienté sans circuit de longueur négative.
Résultat: La matrice A des plus courts chemins.

Initialiser la matrice A ($A=A^{(0)}$).
Pour k variant de 1 à n
Pour i variant de 1 à n
Pour j variant de 1 à n
$a_{ij}=\min(a_{ij}, a_{ik}+a_{kj})$.

Exemple:

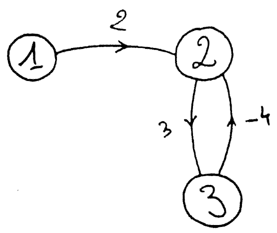


La matrice initiale $A^{(0)}$, les matrices $A^{(1)}$ et $A^{(2)}$, et la matrice des plus courts chemins $A^{(3)}$.

$$A^{(0)}=\begin{pmatrix}0&8&5\\3&0&\infty\\\infty&2&0\end{pmatrix};\ A^{(1)}=\begin{pmatrix}0&8&5\\3&0&8\\\infty&2&0\end{pmatrix};\ A^{(2)}=\begin{pmatrix}0&8&5\\3&0&8\\5&2&0\end{pmatrix};\ A^{(3)}=\begin{pmatrix}0&7&5\\3&0&8\\5&2&0\end{pmatrix}$$

Complexité de l'algorithme $O(n^3)$.

L'algorithme de Floyd dans un graphe avec un circuit de longueur négative.
L'exemple suivant montre comment fonctionne l'algorithme de Floyd dans un graphe avec un circuit de longueur négative.



$$A^{(0)}=\begin{pmatrix}0&2&\infty\\\infty&0&3\\\infty&-4&0\end{pmatrix};\ A^{(1)}=\begin{pmatrix}0&2&\infty\\\infty&0&3\\\infty&-4&0\end{pmatrix}\ A^{(2)}=\begin{pmatrix}0&2&5\\\infty&0&3\\\infty&-4&-1\end{pmatrix};\ A^{(3)}=\begin{pmatrix}0&2&5\\\infty&-1&3\\\infty&-4&-2\end{pmatrix}$$

D'où la modification suivante.

Algorithme Plus court chemin (Floyd) modifié
***Donnée**: Un graphe simple orienté.*
***Résultat**: La matrice A des plus courts chemins ou le message «Circuit détecté»*

Initialiser la matrice A ($A=A^{(0)}$).
Pour k variant de 1 à n
Pour i variant de 1 à n
Si $a_{ii}<0$ alors Arrêter (circuit détecté)
Pour j variant de 1 à n
$a_{ij}=\min(a_{ij}, a_{ik}+a_{kj})$.

V. Problèmes d'ordonnancement.

- Niveau (ou rang) d'un sommet dans un graphe sans circuit.
- Calcul du rang d'un sommet.
- Le problème du plus court chemin dans un graphe sans circuit avec une unique racine.
- Méthode potentiels-tâches
- Méthode PERT

Niveau (ou rang) d'un sommet dans un graphe sans circuit.

Le problème du plus court chemin est facilité dans un graphe sans circuit par la propriété suivante:

Propriété: *Dans un graphe simple orienté sans circuit, il existe toujours un sommet sans prédécesseur.*

Preuve: Considérons G un graphe simple orienté sans circuit dont tous les sommets ont au moins un prédécesseur, soit G* le graphe dual de G, c'est à dire le graphe obtenu à partir de G en inversant le sens de tous les arcs.
On peut dire que G* est aussi sans circuit et que tout sommet de G* a un successeur.
En partant d'un sommet quelconque de G* on peut construire un chemin de longueur arbitrairement longue, comme G* a un nombre fini de sommets, on obtient une contradiction avec le fait que G* est sans circuit.

Dans le cas où le graphe n'a qu'un seul sommet sans prédécesseur x_0, on définit le **rang** d'un sommet i comme étant la cardinalité maximale d'un chemin de x_0 à i.

L'idée de base pour déterminer **le plus court chemin de x_0 à i** est d'examiner tous les chemins de x_0 à i Si on traite chaque sommet i dans l'ordre de son rang, on a la certitude au moment du traitement du sommet i que tous les chemins qui mènent à i ont été examinés

Ci-dessous nous présentons d'abord une méthode de calcul du rang d'un sommet puis le problème du plus court chemin dans un graphe sans circuit avec un seul sommet sans prédécesseur.

Calcul du rang d'un sommet.

La propriété précédente ainsi que le caractère héréditaire du fait d'être sans circuit (tout sous graphe d'un graphe sans circuit est sans circuit) conduisent à un algorithme de calcul du rang des sommets.

Algorithme Rang des sommets.

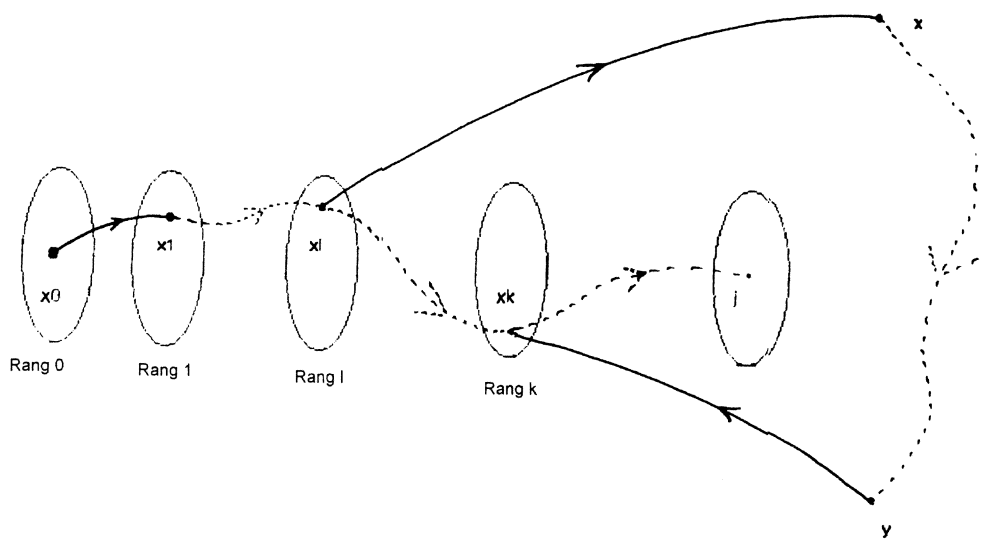
Données: Un graphe simple orienté sans circuit avec un unique sommet sans prédécesseur x_0

Résultat: .Le rang de chaque sommet

Rang courant=0
Tant que le graphe n'est pas vide
    Tous les sommets sans prédécesseur ont le rang courant.
    Retirer du graphe tous les sommets sans prédécesseur.
    Incrémenter le rang courant.

Validité et complexité.

Tout sous graphe d'un graphe sans circuit est sans circuit. Par conséquent dans chaque itération on retire au moins un sommet, la terminaison de l'algorithme est donc assurée.
Observons que les prédécesseurs d'un sommet ont un rang inférieur à ce sommet et que deux sommets de même rang ne peuvent être adjacents.



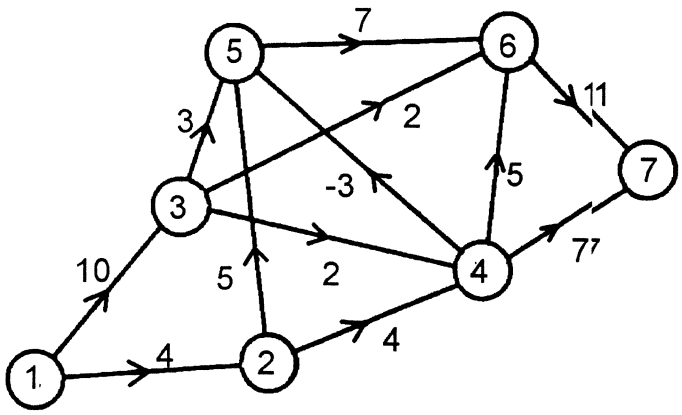
Etant donné un sommet i distinct de x_0, i a un prédécesseur de rang immédiatement inférieur à celui de i.
Donc le sommet i est accessible à partir de x_0 par un chemin qui emprunte exactement un sommet de chaque rang et ce chemin est de cardinalité maximale, en effet s'il existait un chemin plus long de x_0 à i ce chemin contiendrait un sommet x de rang supérieur à i et on aurait donc un sommet avec un prédécesseur y dont le rang lui est supérieur. Contradiction.

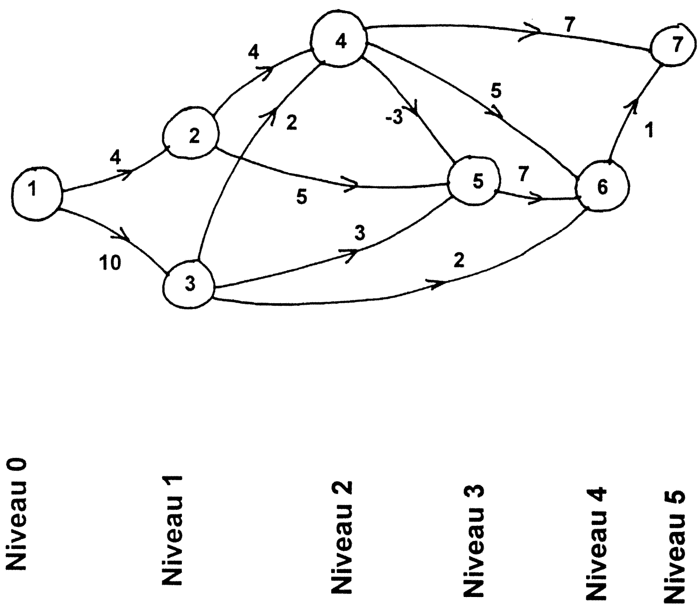
Comme tout sommet est accessible à partir de x_0, x_0 est qualifié de **racine** du graphe.

Complexité O(n+m) avec un graphe donné par listes d'adjacence. O(n^2) pour un graphe donné sous forme matricielle.

Sans alourdir la complexité on peut considérer qu'à l'issue de cet algorithme l'ensemble des sommets est partitionné en **niveaux** (sommets de même rang) et que ces sommets sont triés selon la valeur de leur rang.

Exemple: Un graphe sans circuit avec une unique racine et la partition de ses sommets en niveaux





Le problème du plus court chemin dans un graphe sans circuit avec une unique racine.

On suppose que le rang de chaque sommet a été calculé et que les sommets sont triés dans l'ordre de leur rang.

Algorithme Plus court chemin (Bellman)
Donnée: Un graphe simple orienté sans circuit avec une unique racine x_0.
Résultat: Un vecteur π des plus courts chemins à partir de x_0.

$\pi(x_0)=0$; $\pi(i)=\infty$ pour les autres sommets.
Pour chaque sommet i (dans l'ordre des rangs)
Pour chaque successeur j de i
$\pi(j)=\min(\pi(j),\pi(i)+$ valeur de l'arc (i,j))

Validité et complexité.

Supposons par récurrence sur le rang que pour tout sommet i de rang inférieur strictement à k , $\pi(i)$ est optimal.
Soit j un sommet de rang k et considérons un chemin optimal de x_0 à j, soit i le prédécesseur de j sur ce chemin. **Tout sous chemin d'un chemin optimal étant optimal**, $\pi(i)$ doit être optimal, $\pi(j)$ est donc donné par la formule suivante les prédécesseurs de j étant de rang inférieur à k.

$$\pi(j) = \min_{i \text{ pré dé ceseur de } j}(\pi(i) + \text{valeur de l'arc}(i,j))$$

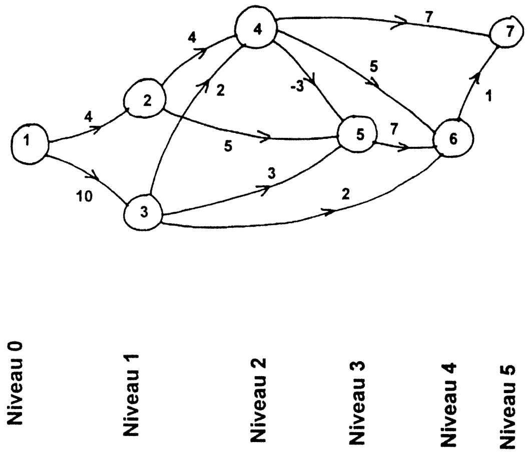
Complexité: O(n+m) pour un graphe donné par listes d'adjacence.

On peut modifier l'algorithme précédent pour avoir le moyen de retrouver les plus courts chemins. On constitue un tableau père tel que père(x) est le prédécesseur de x dans le plus court chemin de x_0 à x, par convention père(x_0)=x_0.

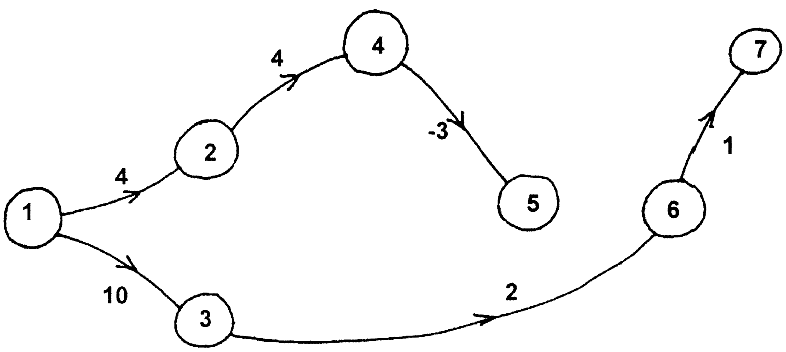
Algorithme Plus court chemin (Bellman) modifié
Donnée: Un graphe simple orienté sans circuit avec une unique racine x_0.
Résultat: Un vecteur π des plus courts chemins à partir de x_0 et l'arborescence des plus courts chemins.

$\pi(x_0)=0$; $\pi(i)=\infty$ pour les autres sommets.
père(x_0)=x_0.
Pour chaque sommet i (dans l'ordre des rangs)
Pour chaque successeur j de i
Si $\pi(j)>\pi(i)+$ valeur de l'arc (i,j)) Alors
père(j)=i.
$\pi(j)=\min(\pi(j),\pi(i)+$ valeur de l'arc (i,j))

Exemple:
Les vecteurs π et père au fur et à mesure des itérations: et l'arborescence des plus courts chemins.



Initialisation: $\pi(0,\infty,\infty,\infty,\infty,\infty,\infty)$; et père(1,X,X,X,X,X,X).
i=1 $\pi(0,4,10,\infty,\infty,\infty,\infty)$; et père(1,1,1,X,X,X,X).
i=3 $\pi(0,4,10,12,13,12,\infty)$; et père(1,1,1,3,3,3,X).
i=2 $\pi(0,4,10,8,9,12,\infty)$; et père(1,1,1,2,2,3,X).
i=4 $\pi(0,4,10,8,5,12,15)$; et père(1,1,1,2,4,3,4).
i=5 $\pi(0,4,10,8,5,12,15)$; et père(1,1,1,2,4,3,4).
i=6 $\pi(0,4,10,8,5,12,13)$; et père(1,1,1,2,4,3,6).
i=7 $\pi(0,4,10,8,5,12,13)$; et père(1,1,1,2,4,3,6).



L'algorithme de Bellmann peut sans difficulté être adapté pour résoudre le problème du plus **long** chemin.

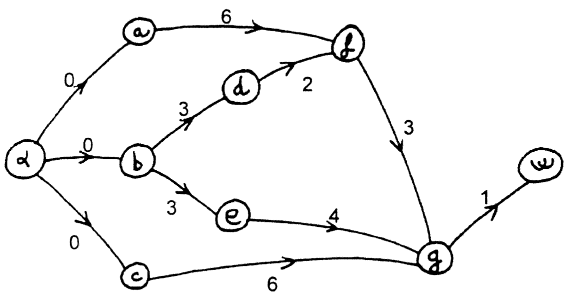
Certaines méthodes pour résoudre des problèmes d'ordonnancement s'inspirent des principes et algorithmes développés ci-dessus, elles sont présentées maintenant.

Méthode potentiels-tâches

(Méthode M.P.M de Roy) 1960[8]

On doit exécuter sept actions a,b,c,d,e,f,g soumises aux impératifs rapportés dans le tableau ci-dessous.

Tâches	Durée	Impératifs
a	6	
b	3	
c	6	
d	2	b doit être terminée
e	4	b doit être terminée.
f	3	d et a doivent être terminées
g	1	f,e,c doivent être terminées



on associe un graphe à ce problème de la façon suivante:
Les sommets sont les tâches à exécuter. les arcs représentent la relation de précédence, les arcs sont valués par la durée de la tâche origine.

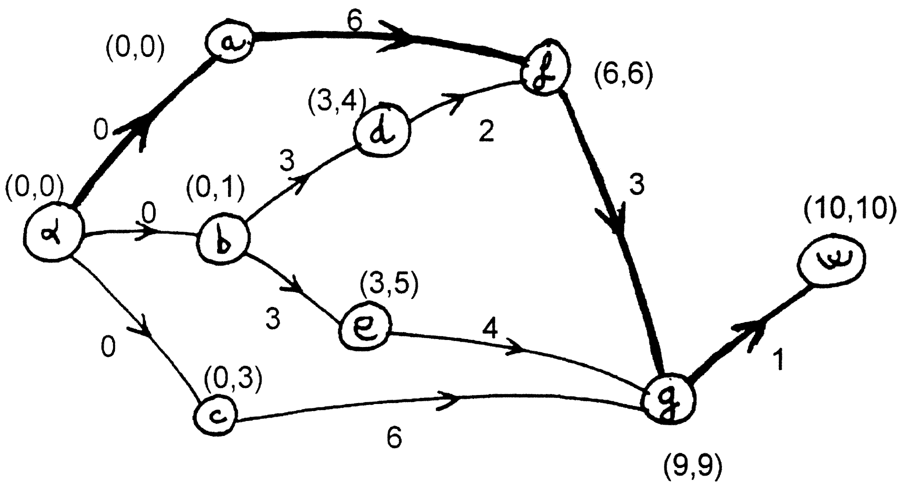
Le graphe obtenu est sans circuit de par la nature de la relation de précédence.
On ajoute une tâche fictive ω pour prendre en compte la durée des dernières tâches et une tâche fictive α pour appliquer l'algorithme de Bellmann.
Toutes les tâches qui précèdent une tâche donnée devant être exécutées avant, on applique l'algorithme de Bellman en calculant le plus **long** chemin à partir de α. Pour chaque tâche on obtient ainsi la **date de début au plus tôt**.
Si on souhaite obtenir la **date de début au plus tard** de chaque tâche sans changer la durée totale d'exécution, on remonte le graphe du dernier niveau vers le premier,
On fixe la date de début au plus tard de la dernière tâche égale à la date de début au plus tôt, puis pour chaque tâche i on calcule la date de début au plus tard de i par la formule:

$$\text{date de début au plus tard (i)} = \min_{\text{j successeur de i}} (\text{date de début au plus tard(j)} - \text{valeur de l'arc(i,j)})$$

Une tâche dont la date de début au plus tard est égale à la date de début au plus tôt est une **tâche critique**. Un chemin formé uniquement de tâches critiques est un

[8] Roseaux (nom collectif), *Exercices et problèmes résolus de recherche opérationnelle, Tome 1 graphes: leurs usages, leurs algorithmes*, Ed MASSON (1986)

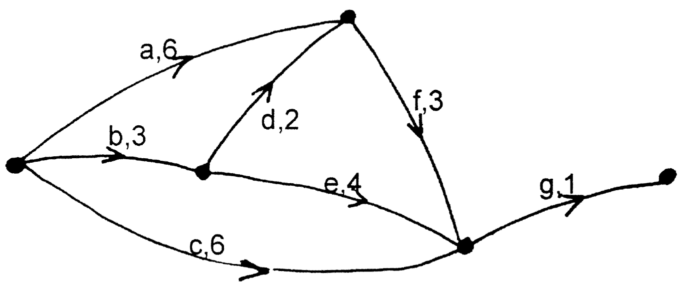
chemin critique, c'est le cas du chemin de la racine à la dernière tâche dans l'arborescence fournie par l'algorithme de Bellman



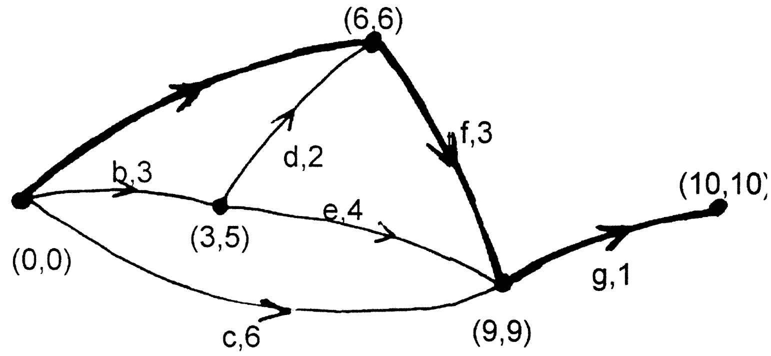
La méthode PERT [9]
(ou méthode potentiels-étapes).

Dans la même situation que précédemment on définit un graphe PERT de la façon suivante: les tâches sont les arcs et les sommets sont les étapes intermédiaires. Pour le construire on peut par exemple considérer une étape de début et une étape de fin pour chaque tâche puis réduire le nombre d'étapes.

Ci-contre, le graphe PERT pour l'exemple précédent.
Le graphe PERT est lui aussi sans circuit.



On procède alors de la même façon que précédemment pour déterminer la date au plus tôt et la date au plus tard de chaque événement.



Un exemple d'utilisation de la méthode P.E.R.T est détaillé dans la deuxième partie (Exercice VII.1)

[9] PERT: Program Evaluation and Research Task.

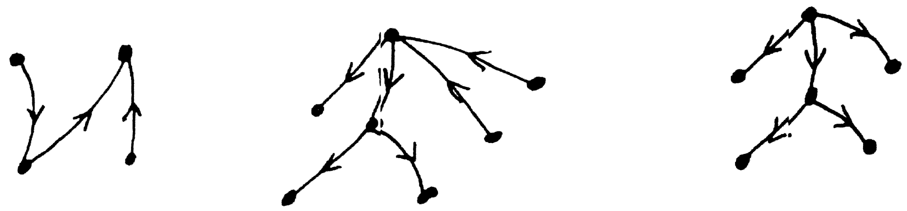
VI. Arbres, arborescence.

Un graphe est **connexe** si et seulement si pour 2 sommets quelconques, il existe une chaîne (éventuellement vide) les reliant.
Cette relation binaire entre les sommets d'un graphe induit une partition de l'ensemble des sommets.
Les classes d'équivalence de cette partition sont les **composantes connexes** du graphe.

Observons qu'un graphe connexe de n sommets possède au moins n-1 arcs (preuve facile par récurrence).

Définition: Un arbre est un graphe simple orienté connexe et sans cycle.

On appelle **forêt** un graphe dont toutes les composantes connexes sont des arbres.
Ci dessous une forêt de 3 arbres dont une arborescence à 6 sommets.



Dans un arbre, un sommet sans successeur est une feuille.

Théorème[10]: Etant donné un graphe simple orienté G de n sommets ($n \geq 2$,) les propriétés suivantes sont équivalentes:
1°) G est un arbre
2°) G est sans cycle et admet n-1 arcs.
3°) G est connexe et admet n-1 arcs.
4°) G est sans cycle et en ajoutant un arc quelconque on crée un cycle.
5°) G est connexe et ne l'est plus si on supprime un arc quelconque.
6°) Tout couple de sommets est relié par une et une seule chaîne.

Preuve: $1°) \Rightarrow 2°) \Rightarrow 3°) \Rightarrow 4°) \Rightarrow 5°) \Rightarrow 6°) \Rightarrow 1°)$
$1°) \Rightarrow 2°)$ Par récurrence sur n. Supposons que G soit sans cycle de n+1 sommets.
Soit x un sommet quelconque de G. On désigne par G-x le graphe G privé de x.
G-x est une forêt de n sommets ayant $C_1,...C_k$ (k est compris entre 1 et n) comme composantes connexes. Chaque composante C_i ayant n_i sommets, d'après l'hypothèse de récurrence, le nombre d'arcs de G-x est .

$$\sum_{i=1}^{k}(n_i - 1) = (\sum_{i=1}^{k} n_i) - k = n - k$$

Par ailleurs pour chaque composante connexe C_i, le graphe C_i+x est connexe sinon G ne le serait pas, donc x a au moins un voisin dans C_i; de plus C_i+x est sans cycle sinon G aurait aussi un cycle, donc x ne peut avoir deux voisins dans C_i.
Par conséquent le nombre d'arcs contenant x est k et le nombre total d'arcs de G est le nombre total d'arcs de G-x augmenté du nombre d'arcs contenant x , c'est à dire de :

[10] Aimé SACHE (nom collectif) *La théorie des graphes* Que sais-je (1974) P.U.F

n-k+k=n.

2°) $\Rightarrow$ 3°) Avec les notations précédentes on a le nombre d'arcs de G égal à n-k où k est le nombre de composantes connexes de G. d'où k=1.

3°) $\Rightarrow$ 4°) même démarche que pour 1°) $\Rightarrow$ 2°). Par récurrence sur le nombre de sommets.

Supposons que G est un graphe connexe de n+1 sommets avec n arcs.

Soit x un sommet quelconque de G. Soit C_1, C_2,..C_k les composantes connexes du graphe G-x (k entre 1 et n).

Chacune de ces C_i comporte n_i sommets et au moins n_i-1 arcs puisqu'elle est connexe.
Comme x appartient à au moins k arcs et que G a n arcs, aucune des composantes C_i n'a plus de n_i-1 arcs. Par conséquent chacune des Ci a exactement n_i-1 arcs et x a exactement un voisin dans chaque composante C_i.
Le reste s'en déduit par récurrence.

4°) $\Rightarrow$ 5°) Si G n'était pas connexe, on pourrait prendre deux sommets x et y dans 2 composantes connexes distinctes et on pourrait ajouter l'arc (x,y) sans créer de cycle. Contradiction.
Si de plus on pouvait retirer un arc de G sans le déconnecter c'est que cet arc faisait partie d'un cycle.

5°) $\Rightarrow$ 6°) L'existence d'une chaîne reliant 2 sommets vient de la connexité. Si 2 sommets étaient reliés par 2 chaînes distinctes, on pourrait choisir de retirer un arc de ces deux chaînes sans déconnecter le graphe.

et 6°) $\Rightarrow$ 1°) Même démarche.

Arborescence.
Si dans un arbre un sommet permet d'accéder à tous les autres ce sommet est la **racine** de l'arbre. La racine d'un arbre si elle existe est unique.

Définition: On appelle arborescence un arbre qui admet une racine.

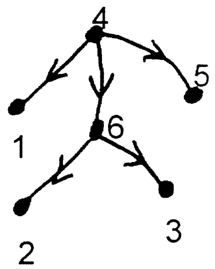
Il résulte de ce qui précède que dans une arborescence:

Il existe un **unique** chemin de la racine à chaque sommet.
Etant donné un sommet x , tout sommet situé sur le chemin de la racine à x (sauf x lui-même) est un **ancêtre** de x.
La cardinalité du plus long chemin de la racine à un sommet est la **hauteur** (ou profondeur de l'arborescence).

Chaque sommet qui n'est pas la racine admet un unique prédécesseur: son **père.**

Cette dernière propriété est d'ailleurs à l'origine d'une méthode de représentation d'une arborescence sous la forme d'un tableau, le tableau «père».
Pour chaque sommet i père(i)= le père de i.
Pour la racine père(racine)=racine.

Exemple:



Sommet	1	2	3	4	5	6
Père	4	6	6	4	4	4

Deuxième partie.

- **Les applications pédagogiques**

I Graphes .

- **Vocabulaire et définitions.**
- **Représentations d'un graphe en machine.**

Vocabulaire et définitions.

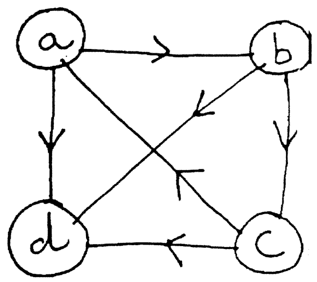
* **Exercice I.1.** Le graphe (X,U) défini ci dessous représente les résultats d'un tournoi de tennis.
X={a,b,c,d} est l'ensemble des joueurs
U={(a,d),(a,b),(b,d),(b,c),(c,d),(c,a)}. est l'ensemble des arcs.

L'existence de l'arc (i,j) signifie que le joueur i a remporté l'unique partie l'opposant au joueur j.

1°) Donner une représentation géometrique du graphe.
Définir les fonctions successeur et prédécesseur Γ et Γ^{-1}
Que signifie concrètement $y \in \Gamma(x)$ et $y \in \Gamma^{-1}(x)$?
Définir et écrire la matrice d'adjacence du graphe.

2°) Donner pour chaque sommet x, le nombre de successeurs, (degré sortant de x), le nombre de prédécesseurs (degré entrant), puis le nombre de voisins. (degré de x) Quelles en sont les significations concrètes ?

Corrigé :1°)



x	$\Gamma(x)$	$\Gamma^{-1}(x)$	$\Gamma(x) U \Gamma^{-1}(x)$
a	{b,d}	{c}	{b,c,d}
b	{c,d}	{a}	{a,c,d}
c	{a,d}	{b}	{a,b,d}
d	∅	{a,b,c}	{a,b,c}

$y \in \Gamma(x) \Leftrightarrow (x, y) \in U$, signifie : x a gagné la partie l'opposant à y.

$y \in \Gamma^{-1}(x) \Leftrightarrow (y, x) \in U$ signifie y a gagné la partie l'opposant à x.

$\Gamma(x) \cup \Gamma^{-1}(x)$ est l'ensemble des voisins de x, c'est à dire l'ensemble des adversaires de x.

Nous voyons ainsi que dans ce tournoi, chaque concurrent a joué avec les trois autres (tournoi régulier).

La matrice d'adjacence du graphe :

$$M = (a_{ij}) \begin{cases} a_{ij} = 1 \text{ si } (i,j) \in U \\ a_{ij} = 0 \text{ sinon} \end{cases} \text{ ou } \begin{cases} a_{ij} = 1 \text{ si } j \in \Gamma(i) \\ a_{ij} = 0 \text{ sinon} \end{cases}$$

La matrice d'adjacence

$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

2°)

x	*d⁺(x)*	*d⁻(x)*	*d(x)*
a	*2*	*1*	*3*
b	*2*	*1*	*3*
c	*2*	*1*	*3*
d	*0*	*3*	*3*

$d^+(x)$ représente le nombre de parties gagnées par le joueur.x
$d^-(x)$ représente le nombre de parties perdues par le joueur x.
$d(x)$ représente le nombre de parties jouées par x.

Remarquons que :

$d^+(x) = \mathrm{Card}(\Gamma(x)) = \sum_j a_{ij}$; $d^-(x) = \mathrm{Card}(\Gamma^{-1}(x)) = \sum_i a_{ij}$; $d(x) = \mathrm{Card}(\Gamma(x) \cup \Gamma^{-1}(x))$

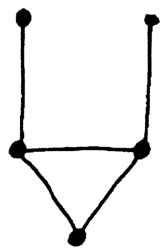
$d(x) = d^+(x) + d^-(x)$.

Exercice I.2. Quelle est la longueur maximale d'un chemin élémentaire dans un graphe orienté non valué de n sommets?
Corrigé: c'est n-1, ce chemin est alors hamiltonien.

****Exercice I.3 :** 1°)Existe-t-il un graphe simple non orienté dont la suite des degrés est :
(1,1,2,3,3) , c'est à dire dont les sommets ont respectivement pour degré 1,1,2,3,3 ?

2°) b) Démontrer qu'il n'existe pas de graphe simple non orienté connexe sans boucle ayant 8 sommets dont les degrés des sommets sont : 1,1,1,2,3,4,5, et 7?

Corrigé : 1°)«Un seul» graphe correspond à cette séquence : le «taureau»



2°) Si on passe au graphe complémentaire (définition page suivante), un sommet est isolé.
Le problème peut donc être reformulé comme suit :
Démontrer qu'il n'existe pas de graphe simple non orienté connexe sans boucle ayant 7 sommets dont les degrés des sommets sont 5,5,5,4,3,2,1?"

Considérons a,b,c,d,e,f,g les sommets d'un tel graphe rangés dans l'ordre décroissant du degré : 5,5,5,4,3,2,1.
Observons d'abord que le sommet g (celui de degré 1) doit être adjacent à a ou b ou c, sinon les trois sommets de degré 5 devraient être adjacents à d,e,f et on n'aurait plus alors de sommet de degré 2.
Supposons désormais que g est adjacent à a.
On a alors :
g est non adjacent à b ni à c,
b et c sont donc adjacents à tous les sommets sauf g.
Le sommet f étant de degré 2 ses seuls voisins sont b et c. Par conséquent les voisins de a sont b,g,d,e,c
Les sommets d et e sont donc respectivement de degré 4 ou 3 selon qu'il sont adjacents ou non.

Commentaires : On sait très peu de choses sur le degré d'un graphe,
Le problème de savoir quelles classes de graphes peuvent être caractérisées par la suite des degrés de leurs sommets reste ouvert.
Plus généralement le problème de l'isomorphisme est ouvert lui aussi. (2 graphes sont isomorphes s'il existe une bijection préservant les adjacences entre les ensembles de sommets).

***Exercice I.4**: **Graphe dual et complémentaire d'un graphe.**
Etant donné un graphe simple orienté G on définit le graphe dual de G noté G* en inversant les arcs de G et on définit le complémentaire de G noté G^c ou $\overline{G}$ en ajoutant les arcs absents dans G et en retirant les arcs présents dans G.
Plus précisément:
(i,,j) est un arc de G^* si et seulement si (j, i) est un arc de G.
(i,,j) est un arc de $\overline{G}$ si et seulement si (i, ,j), n'est pas un arc de G.
1°) Calculer $(G^*)^*$ et $\overline{\overline{G}}$.
2°) Caractériser les graphes vérifiant

a) $G^*=G$.
b) Ceux vérifiant $\overline{G}=G$.
c) Et ceux vérifiant $G^*=\overline{G}$.

3°) Donner des algorithmes matriciels permettant de calculer le graphe dual et le complémentaire d'un graphe G.

Corrigé :
2°)a) Les graphes symétriques (non orientés).
2°)b) Les graphes sans arc et n'ayant qu'un sommet et le graphe vide.
Si un graphe G vérifie $\overline{G}=G$ alors tout sous graphe induit de G vérifie la même propriété. Or aucun graphe de 2 sommets ne vérifie $\overline{G}=G$. En conséquence G ne peut avoir 2 sommets ou plus, la conclusion suit immédiatement.
2°)c) Les graphes tels que 2 sommets distincts quelconques sont reliés par un arc unique. (ce sont les graphes totaux asymétriques sans boucles autrement dit tournois).
3°) A partir de la matrice booléenne d'adjacence, on calcule le dual en transposant la matrice et on calcule le complément en faisant un NON booléen sur chaque coefficient

***Exercice :I.5** [11] Montrer que dans tout groupe d'au moins 2 personnes il y en a au moins 2 qui ont le même nombre d'amis dans ce groupe.(la relation «est l'ami de» est symétrique et anti-réflexive)

Solution: On modélise la situation par un graphe non orienté où les sommets représentent les personnes et les arêtes représentent la relation «est l'ami de». Dire que deux personnes quelconques n'ont pas le même nombre d'amis revient à dire qu'il n'y a pas deux sommets du graphe ayant le même degré. Si le graphe a n sommets il y a donc n valeurs différentes de degré.Dans le cas où il existe un sommet de degré 0 on achève par réccurence,dans le cas contraire Il y a donc n valeurs strictement positives différentes de degré. Il y a donc un sommet de degré n. Or ceci est impossible, dans un graphe non orienté sans boucle de n sommets, le plus grand degré possible est n-1.

[11] J.A.Bondy and U.S.R.Murty *Graph Theory with Applications*, North-Holland, New York, (1979)

Représentations d'un graphe en machine.

***Exercice I.6:** Pour chacune des représentations d'un graphe orienté en machine tableau des successeurs, matrice sommets-arcs ou matrice d'adjacence, donner un algorithme réalisant les opérations élémentaires: test de l'existence d'un arc, calcul du degré sortant d'un sommet, suppression d'un arc.

Corrigé:

Opération	*Tableau des successeurs (listes d'adjacence) chaque sommet a une liste de successeurs*
Test de l'existence d'un arc. ***Algorithme*** *Existence d'un arc.* ***Donnée:*** *un couple de sommets (i,j)* ***Résultat:*** *Oui ou Non selon que l'arc (i,j) existe ou pas.*	*Pour chaque successeur s de i* *Si s=j retourner OUI.* *Retourner NON.*
Calcul du degré sortant d'un sommet ***Algorithme*** *Degré sortant d'un sommet.* ***Donnée:*** *un sommet i* ***Résultat:*** *le nombre de successeurs de i.*	*Initialiser le degré de i à 0.* *Pour chaque successeur de i* *Ajouter 1 au degré de i.*
Suppression d'un arc (i,j) ***Algorithme*** *Suppression d'un arc.* ***Donnée:*** *un arc (i,j)* ***Sortie:*** *le graphe sans l'arc (i,j).*	***Cas orienté:*** *Pour chaque successeur s de i* *Si s=j alors supprimer s de la liste des successeurs de i.* ***Cas non orienté:*** *Pour chaque successeur s de i* *Si s=j alors supprimer s de la liste des successeurs de i.* *Pour chaque successeur s de j* *Si s=i alors supprimer s de la liste des successeurs de j.*

Opération	*Matrice sommets-arcs. Le graphe est donné sous la forme d'une matrice A de taille nxm*
Test de l'existence d'un arc. ***Algorithme*** *Existence d'un arc.* ***Donnée:*** *un couple de sommets (i,j)* ***Résultat:*** *Oui ou Non selon que l'arc (i,j) existe ou pas.*	*Pour chaque colonne c de la matrice A* *Si A[i,c]= -1 et A[j,c]=1 retourner OUI.* *Retourner NON.*
Calcul du degré sortant d'un sommet ***Algorithme*** *Degré sortant d'un sommet.* ***Donnée:*** *un sommet i* ***Résultat:*** *le nombre de successeurs de i.*	*Initialiser le degré de i à 0.* *Pour chaque colonne de la matrice A* *Si A[i,c]=-1 alors Ajouter 1 au degré de i.*
Suppression d'un arc (i,j) ***Algorithme*** *Suppression d'un arc.* ***Donnée:*** *un arc (i,j)* ***Résultat:*** *le graphe sans l'arc (i,j).*	*Pour chaque colonne c de la matrice A* *Si A[i,c]= -1 et A[j,c]=1 alors faire* *A[i,c]=A[j,c]=0*

Opération	*Matrice sommet-sommet.* *Le graphe est donné sous la forme d'une matrice booléenne A de taille nxn*
Test de l'existence d'un arc. ***Algorithme** Existence d'un arc.* ***Donnée:** un couple de sommets (i,j)* ***Résultat**: Oui ou Non selon que l'arc (i,j) existe ou pas.*	*Si A[i,j]=1 retourner OUI.sinon retourner NON.*
Calcul du degré sortant d'un sommet ***Algorithme** Degré sortant d'un sommet.* ***Donnée:** un sommet i* ***Résultat**:le nombre de successeurs de i.*	*Initialiser le degré de i à 0.* *Pour chaque colonne de la matrice A* *Si A[i,c]= 1 alors Ajouter 1 au degré de i.*
Suppression d'un arc (i,j) ***Algorithme** Suppression d'un arc.* ***Donnée:** un arc (i,j)* ***Résultat**:le graphe sans l'arc (i,j).*	*A[i,j]=0*

***Exercice I.7: Des algorithmes de passage d'une implantation à une autre pour un graphe simple orienté.**

Corrigé:

	Matrice sommet-sommet *le graphe est stocké sous la forme d'une matrice booléenne A de taille nxn*	***Matrice sommet-arc*** *le graphe est stocké sous la forme d'une matrice B de taille nxm*
Liste d'adjacence *chaque sommet a sa liste de successeurs*	*Pour i variant de 1 à n* *Pour j variant de 1 à n* *A[i,j]=0.* *Pour chaque sommet x du graphe* *Pour chaque successeur s de x* *A[x,s]=1.*	*Pour i variant de 1 à n* *Pour j variant de 1 à m* *B[i,j]=0.* *numéro d'arc=0.* *Pour chaque sommet x du graphe* *Pour chaque successeur s de x* *Incrémenter le numéro d'arc.* *B[x,numéro d'arc]=-1.* *B[s, numéro d'arc]=+1*

	Liste d'adjacence *chaque sommet a sa liste de successeurs*	***Matrice sommet-arc*** *le graphe est stocké sous la forme d'une matrice B de taille nxm*
Matrice sommet-sommet *le graphe est stocké sous la forme d'une matrice booléenne A de taille nxn*	*Préparer un tableau de n sommets avec des listes de successeurs vides.* *Pour i variant de 1 à n* *Pour j variant de 1 à n* *Si A[i,j]=1 alors* *Ajouter j à la liste des successeurs de i*	*Pour i variant de 1 à n* *Pour j variant de 1 à m* *B[i,j]=0.* *numéro d'arc=0.* *Pour i variant de 1 à n* *Pour j variant de 1 à n* *Si A[i,j]=1 alors* *Incrémenter le numéro d'arc.* *B[i,numéro d'arc]=-1.* *B[j, numéro d'arc]=+1*

	Liste d'adjacence *chaque sommet a sa liste de successeurs*	***Matrice sommet-sommet*** *le graphe est stocké sous la forme d'une matrice booléenne A de taille nxn*
Matrice sommet-arc *le graphe est stocké sous la forme d'une matrice B de taille nxm*	*Préparer un tableau de n sommets avec des listes de successeurs vides.* *Pour chaque colonne c de la matrice B* *Pour chaque ligne l de la matrice B* *Si B[l,c]= -1 alors origine=l* *Si B[l,c]= +1 alors extrémité=l.* *Ajouter le sommet extrémité comme successeur du sommet origine.*	*Pour i variant de 1 à n* *Pour j variant de 1 à n* *A[i,j]=0.* *Pour chaque colonne c de la matrice B* *Pour chaque ligne l de la matrice B* *Si B[l,c]= -1 alors origine=l* *Si B[l,c]= +1 alors extrémité=l.* *A[origine, extrémité]=1.*

**Thème d'étude I.8: Une représentation compacte d'un graphe en machine.

A l'époque où la mémoire était coûteuse, l'optimisation du stockage de l'information était de rigueur . Voici un exemple de représentation d'un graphe en machine n'utilisant que deux tableaux α et β.
L'idée de base est la suivante : on juxtapose dans un unique tableau β les différentes listes de successeurs.
Ainsi dans l'exemple ci –dessous le tableau des successeurs

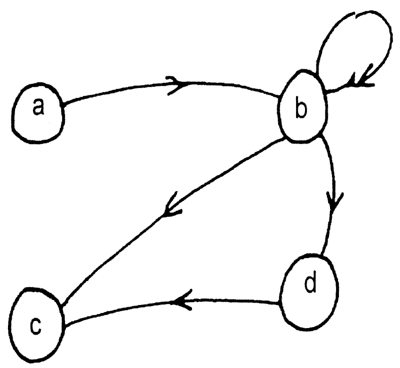
Sommets	Successeurs
a	b
b	b,c,d
c	∅
d	c

est compacté en

indice	1	2	3	4	5
β	b	b	c	d	c

Il reste à savoir situer, pour chaque sommet, l'endroit où débute et l'endroit où se termine la liste de ses successeurs, c'est le rôle du tableau α qui donne pour chaque sommet l'adresse du début de la liste des successeurs dans le tableau β
Plus précisément :
Le tableau α est indexé par les sommets du graphe, le second tableau β est formé d'une juxtaposition des listes de successeurs. Etant donné un sommet i, α[i] donne l'indice dans le tableau β du premier successeur de i s'il existe. Ainsi le premier successeur du sommet i s'il existe, est donné par $\beta[\alpha[i]]$ tandis que le dernier successeur de i s'il existe, est donné par $\beta[\alpha[i+1]-1]$. Un sommet fictif n+1 est ajouté, par convention $\beta[\alpha[n+1]]=0$ pour marquer l'arrêt de la dernière liste de successeurs.

Exemple: Dans le but de clarifier les choses nous donnons ici deux représentations différentes du **même** graphe à l'aide de tableaux α et β



1ère **représentation** : Dans le tableau β ,les listes sont juxtaposées dans l'ordre :
Successeurs de a, Successeurs de b, Successeurs de c (vide), Successeurs de d :

Sommet	a	b	c	d	5
α	1	2	5	5	6

indice	1	2	3	4	5	6
β	b	b	c	d	c	0

2ème **représentation** : Dans le tableau β ,les listes sont juxtaposées dans l'ordre :
Successeurs de d, Successeurs de c (vide), Successeurs de b, Successeurs de a :

sommet	d	c	b	a	5
α	1	2	2	5	6

indice	1	2	3	4	5	6
β	c	b	c	d	b	0

On peut alors envisager de :

1°) Donner des algorithmes et leur complexité pour les opérations sur les graphes qui sont:

Le test de l'existence de l'arc (i,j).
Le calcul du degré entrant d'un sommet i.
La suppression de l'arc (i,j).

2°) Donner des algorithmes et leur complexité pour faire le passage d'une représentation α β à une autre (liste d'adjacence ou matrice).

On peut aussi envisager une représentation en tableaux α' et β' où l'on juxtapose les liste de prédécesseurs dans le tableau β' au lieu de listes de successeurs et donner un algorithme permettant de faire le passage d'une représentation α β avec des listes de successeurs à une représentation α' β' avec des listes de prédécesseurs.

Solution: Passage d'une représentation de type α β à une représentation du type $\alpha'\beta'$

a) Construire un tableau «arc» des arcs du graphe en parcourant les tableaux α et β.

```
k=0
Pour chaque sommet i du graphe
        Pour j=α[i] à α[i+1]-1
                Incrémenter k
                arc[k]=(i,β[j])
```

Avec l'exemple précédent on aurait:

indice	*1*	*2*	*3*	*4*	*5*
arc	*(a,b)*	*(b,b)*	*(b,c)*	*(b,d)*	*(d,c)*

b) Trier le tableau arc selon la deuxième coordonnée.

peut se faire en O(m) où m est le nombre d'arcs.

Avec l'exemple précédent on aurait:

indice	*1*	*2*	*3*	*4*	*5*
arc	*(a,b)*	*(b,b)*	*(b,c)*	*(d,c)*	*(b,d)*

c) Reconstruire les tableaux α' et β' par un parcours du tableau arc.

```
sommet courant=0.
index β'=1.
Pour chaque arc (i,j) du tableau arc
    Tant que j n'est pas le sommet courant
        Incrémenter le sommet courant
        α'[sommet courant]=index β'.
    β'[index β']=i.
    Incrémenter index β'
α'[sommet courant+1]=index β'
β'[index β']=0.
```

Avec l'exemple précédent on obtient:

Sommet	*a*	*b*	*c*	*d*	*5*
α'	*1*	*1*	*3*	*5*	*6*

index β'	*1*	*2*	*3*	*4*	*5*	*6*
β'	*a*	*b*	*b*	*d*	*b*	*0*

Coût de la transformation: O(n+m).

II Chemins de longueur p .

- **Leur dénombrement. Cas des multigraphes.**
- **Comment exhiber un chemin ?.**

Thème d'étude

** II.1 Dénombrement des chemins de longueur p dans un graphe simple orienté.

Le calcul de la puissance p ème de la matrice booléenne d'adjacence d'un graphe simple orienté renseigne sur l'existence d'un chemin de longueur p. Comment faire pour dénombrer ces chemins ?
Solution : Il suffit en fait de considérer la matrice d'adjacence comme étant à coefficients réels et de faire les calculs dans (R,+,x) plutôt que dans ({0,1}, ou, et).

On peut alors démontrer de la même façon que dans la première partie la propriété suivante :
Propriété: *Soit G un graphe simple orienté donné par sa matrice d'adjacence A et soit p un nombre entier naturel.*
On désigne par $a^{(p)}_{ij}$ *le coefficient de la ième ligne et de la jème colonne de la matrice* A^p*.(la puissance p ème de A quand la multiplication matricielle est celle induite par les opérations addition et multiplication sur les réels)*

On a alors $a^{(p)}_{ij}$ *qui est le nombre de chemins de longueur p du sommet i vers le sommet j.*

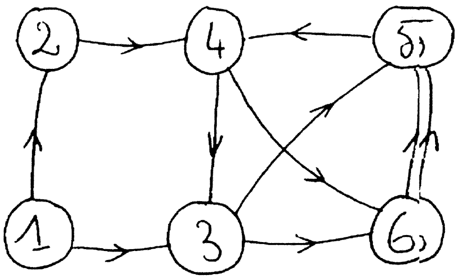
Preuve par réccurence sur p.
La propriété est vraie pour p=1 puisque $a_{ij}=1$ *exprime qu'il existe un arc (i,j) et donc qu'il existe un chemin de longueur 1 de i vers j.*
Supposons que la propriété est vraie au rang p-1.

$A^p = A^{p-1}A \Leftrightarrow a_{ij}^{(p)} = \sum_{k=1}^{n} a_{ik}^{(p-1)} a_{kj}$ *(définition du produit de deux matrices)*

Par hypothèse $a_{ik}^{(p-1)}$ *est le nombre de chemins de longueur p-1 de i à k et* $a_{kj} = 1$ *s'il existe un arc de k à j et 0 sinon.*
Donc $a_{ik}^{(p-1)} \times a_{kj}$ *est le nombre de chemins de longueur p de i à j ayant k comme avant-dernier sommet.*
Ainsi $\sum_{k=1}^{n} a_{ik}^{(p-1)} \times a_{kj}$ *est le nombre total de chemins de longueur p de i à j.*

On peut facilement étendre ce résultat au cas des multigraphes, le coefficient a_{ij} de la matrice d'adjacence représente alors le nombre d'arcs de i vers j.
Les deux exercices qui suivent traitent du dénombrement des chemins de longueur p, le premier dans un graphe simple, le second dans un multigraphe.

* **Exercice II.2**. On considère le graphe suivant:



1°) A étant la matrice d'adjacence du graphe, calculer matriciellement A^p pour p=5 et p=6, à l'aide de la calculatrice.

Donner par exemple le nombre de chemins de longueur 5 du sommet 1 au sommet 5 et définir ces chemins en se référent à la représentation géométrique.

2°) Le graphe contient-il des circuits de longueur 6?

3°) Soit G un graphe simple orienté comportant n sommets avec la matrice A comme matrice d'adjacence. Démontrer la propriété générale suivante:
Si A^n n'est pas nulle alors le graphe G contient des circuits

Corrigé :
1°)

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix};\ A^5 = \begin{pmatrix} 0 & 0 & 1 & 2 & 3 & 2 \\ 0 & 0 & 2 & 1 & 0 & 2 \\ 0 & 0 & 0 & 2 & 3 & 1 \\ 0 & 0 & 1 & 0 & 4 & 3 \\ 0 & 0 & 2 & 1 & 0 & 2 \\ 0 & 0 & 0 & 2 & 1 & 0 \end{pmatrix};\ A^6 = \begin{pmatrix} 0 & 0 & 2 & 3 & 3 & 3 \\ 0 & 0 & 1 & 0 & 4 & 3 \\ 0 & 0 & 2 & 3 & 1 & 2 \\ 0 & 0 & 0 & 4 & 4 & 1 \\ 0 & 0 & 1 & 0 & 4 & 3 \\ 0 & 0 & 2 & 1 & 0 & 2 \end{pmatrix}$$

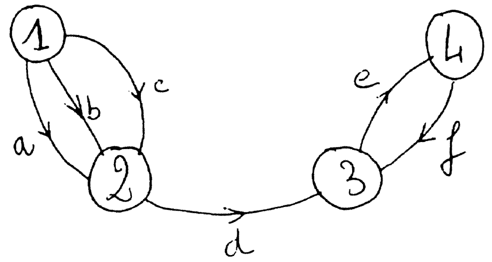
La matrice A^5 donne le nombre de chemins de longueur 5 de i à j.
Par exemple il existe 3 chemins de longueur 5 du sommet 1 au sommet 5 qui sont :
(1,2,4,3,6,5), (1,3,5,4,6,5) (1,3,5,4,3,5)

2°) Le graphe contient de nombreux circuits, dans A^6 on voit que a_{33}=2. Il existe donc 2 chemins de longueur 6 du sommet 3 au sommet 3 c'est à dire 2 circuits de longueur 6. On remarque de même que les termes a_{44}, a_{55}, a_{66} ne sont pas nuls, ce qui dénote la présence de 4,5,6 sur un circuit de longueur 6.

3°) Tout chemin de longueur n dans un graphe simple à n sommets passe au moins 2 fois par le même sommet (la longueur maximale d'un chemin élémentaire est n-1).
Si A^n n'est pas nulle c'est qu'il existe un chemin de longueur n et donc un circuit.
Observons que la réciproque est vraie :
S'il existe un circuit dans un graphe, on peut alors trouver un chemin de longueur arbitraire dans le graphe c'est donc qu'il existe un chemin de longueur n et donc que A^n n'est pas nulle.

La propriété A^n *=0 peut donc caractériser les graphes sans circuit.*

* **Exercice II.3**. On considère le mutigraphe suivant.



1°) A partir de la représentation géométrique du graphe, déterminer la matrice traduisant le nombre de chemins de longueur p entre tout couple (i,j) de sommets pour p=1 (matrice M) puis pour p=2,3,4,5

2°) Calculer matriciellement M^p pour p=1,2,3,4,5

3°) Conclure.

Corrigé :

$$\text{pour } p=1; M=\begin{pmatrix}0&3&0&0\\0&0&1&0\\0&0&0&1\\0&0&1&0\end{pmatrix};\ \text{pour } p=2;\begin{pmatrix}0&0&3&0\\0&0&0&1\\0&0&1&0\\0&0&0&1\end{pmatrix};\ \text{pour } p=3;\begin{pmatrix}0&0&0&3\\0&0&1&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$$

$$\text{pour } p=4;\begin{pmatrix}0&0&0&0\\0&0&0&1\\0&0&1&0\\0&0&0&1\end{pmatrix};\ \text{pour } p=5;\begin{pmatrix}0&0&0&3\\0&0&1&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$$

2°) On retrouve les mêmes résultats par le calcul matriciel.

3°) On peut conclure que le nombre de chemins de longueur p est donné par M^p *avec* $M=(a_{ij})$ *,* a_{ij} *étant le nombre d'arcs (chemins de longueur 1) de i à j.*

***Thème d'étude II.4: Exhiber un chemin de longueur p.

Le calcul de la puissance p ème de la matrice d'adjacence d'un graphe simple orienté renseigne sur l'existence d'un chemin de longueur p. Comment faire pour exhiber ce chemin ?

Solution :

Cela demande de pénétrer au cœur du calcul matriciel booléen : $A^p = A^{p-1} x A$.

Au moment où le coefficient $a_{ij}^{(p)}$ *de la matrice* A^p *est calculé à partir des coefficients de* A^{p-1} *et de A, on peut savoir exactement quel terme parmi les* $a_{ik}^{(p-1)} \times a_{kj}$ *fait passer* $a_{ij}^{(p)}$ *à 1, on sait alors qu'il existe un chemin de longueur p-1 entre i et k auquel s'ajoute un chemin de longueur 1 entre k et j pour former un chemin de longueur p. Il suffit de mémoriser cela au passage dans une structure adaptée.*

Nous donnons d'abord une solution algorithmique détaillée pour l'existence d'un chemin de longueur p, puis une solution algorithmique pour exhiber un chemin de longueur p.

Algorithme Produit matriciel.
Données: Deux matrices carrées booléennes d'ordre n $A=(a_{ij})$ et $B=(b_{ij})$
Résultat: La matrice produit $C=(c_{ij})=AxB$.

```
Pour i variant de 1 à n
        Pour j variant de 1 à n
                Cij=0 ;
                Pour k variant de 1 à n
                        cij =cij +aik x bkj
```

Algorithme Chemin de longueur p.
Données: Un graphe simple orienté donné par sa matrice d'adjacence $A=(a_{ij})$
et un nombre entier naturel p.
Résultat: La matrice A^p

```
M=A
Pour l variant de 2 à p
        M=MxA
Renvoyer M comme résultat.
```

Complexité : $O(n^3)$ *pour p fixé.*

La structure utilisée pour mémoriser les chemins de longueur p est un tableau à 2 entrées (une matrice) $T^{(p)}=(t^{(p)}_{ij})$ où $t^{(p)}_{ij}$ est un chemin de longueur p de i à j

Algorithme Produit matriciel (modifié).

Données: Deux matrices carrées booléennes d'ordre n $A=(a_{ij})$ et $B=(b_{ij})$ et le tableau $T^{(l)}$

Résultat: La matrice produit $C=(c_{ij})=AxB$. Et le tableau $T^{(l+1)}$

```
Pour i variant de 1 à n
      Pour j variant de 1 à n
            Cij=0 ;
            Pour k variant de 1 à n
                  Si aik x bkj =1 alors
                        cij =cij +aik x bkj
                        Insérer le sommet k à la fin de t^l_ij pour former t^(l+1)_ij.
```

Algorithme Exhiber un chemin de longueur p.

Données: Un graphe simple orienté donné par sa matrice d'adjacence $A=(a_{ij})$ et un nombre entier naturel p.

Résultat: La matrice A^p et le tableau $T^{(p)}$

```
M=A ; t^(1)_ij= l'arc (i,j) s'il existe.
Pour l variant de 2 à p
      M=MxA
Retourner M et T^(p)
```

Complexité : $O(n^3)$ pour p fixé.

Prolongement possible : Exhiber ***tous*** *les chemins de longueur p.*

III Fermeture transitive :

- **Exercices**
- **Liaisons entre graphes et calcul matriciel booléen.**
  - **TD1.**
  - **TD2.**
- **Thème d'étude : Algorithme de Warshall**

Exercices.

Exercice III.1: Comment détecter un circuit dans un graphe orienté en calculant la fermeture transitive?

Solution:
Un sommet x se trouve sur un circuit si et seulement si il est accessible à partir de lui même.
Si et seulement si il y a une boucle sur x dans la fermeture transitive.

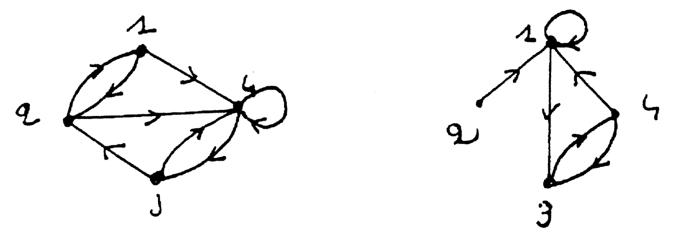
Il suffit par exemple de tester l'existence du coefficient 1 dans la première diagonale de la matrice de la fermeture transitive. Le sommet correspondant se trouve sur un circuit.
Complexité du calcul: celle du calcul de la fermeture transitive.

Exercice III.2:
1°) Quelle est la fermeture transitive d'un circuit?
2°) Quelle est la fermeture transitive d'un graphe simple non orienté?

Réponse
1°) un graphe complet (qui possède tous les arcs possibles)
2°)Des composantes connexes complètes (2 sommets d'une même composante connexe sont toujours mutuellement accessibles dans un graphe non orienté).

* TD1 III.3.

Soient G_1 et G_2 les deux graphes représentés ci dessous.



1°) Définir les fonctions successeur Γ_1 et Γ_2 des graphes G_1 et G_2 puis les matrices booléennes M_1 et M_2 de ces deux graphes.

2°) Définir dans un tableau les fonctions $\Gamma_1 \cup \Gamma_2$ puis, $\Gamma_2 \bullet \Gamma_1$ applications de X dans l'ensemble des parties de X telles que :

$$(\Gamma_1 U \Gamma_2)(x) = \Gamma_1(x) U \Gamma_2(x)$$
$$(\Gamma_2 \bullet \Gamma_1)(x) = \Gamma_2(\Gamma_1(x))$$

En donner la signification.
Ecrire la matrice des graphes $G=(X,\Gamma_1 U \Gamma_2)$ et $G'=(X,\Gamma_2 \bullet \Gamma_1)$

3°) Calculer les matrices booléennes $M_1 + M_2$ et $M_1 \bullet M_2$.

4°) Conclure.

5°) Représenter géométriquement les graphes $G=(X, \Gamma_1 \cup \Gamma_2)$ et $G'=(X, \Gamma_2 \bullet \Gamma_1)$.

Corrigé :

1°) X={1,2,3,4}.

X	$\Gamma_1(x)$
1	{2,4}
2	{1,4}
3	{2,4}
4	{3,4}

X	$\Gamma_2(x)$
1	{1,3}
2	{1}
3	{4}
4	{1,3}

$$M_1 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}; \quad M_2 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

X	$\Gamma_1 \cup \Gamma_2$ (x)
1	{1,2,3,4}
2	{1,4}
3	{2,4}
4	{1,3,4}

X	$\Gamma_2 \bullet \Gamma_1$ (x)
1	{1,3}
2	{1,3}
3	{1;3}
4	{1,3,4}

2°)

Ex : $(\Gamma_1 U \Gamma_2)(1)=\{2,4\}U\{1,3\}=\{1,2,3,4\}]$ $\quad (\Gamma_2 \bullet \Gamma_1)(1)=\Gamma_2(\{2,4\})=\{1,3\}$.

$(\Gamma_1 U \Gamma_2)(x)$ *est l'ensemble des sommets atteints en parcourant à partir de x un arc de* G_1 *ou un arc de* G_2.
$(\Gamma_2 \Gamma_1)(x)$ *est l'ensemble des extrémités atteintes en parcourant à partir de x un arc de* G_1 *puis à partir des extrémités précédentes un arc de* G_2.

Matrice de $G = (X, \Gamma_1 \cup \Gamma_2)$

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

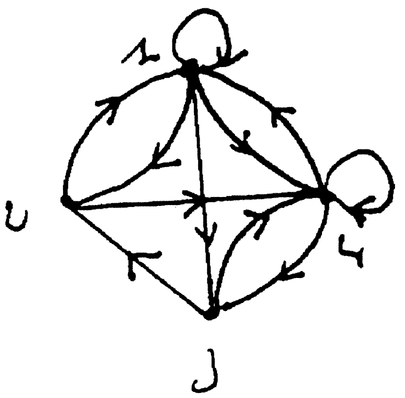
Matrice de $G' = (X, \Gamma_2 \bullet \Gamma_1)$

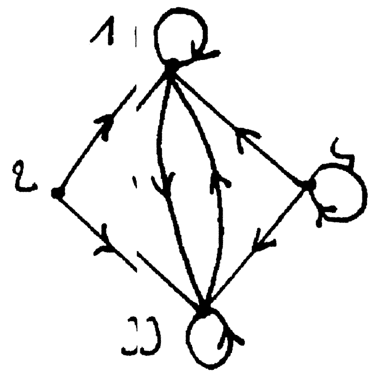
$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

3°) Calcul des matrices booléennes M_1+M_2 *et* $M_1.M_2$.
On retrouve les matrices précédemment écrites.

4°) Concluons $M_1 + M_2$ *est la matrice d'adjacence du graphe* $G=(X, \Gamma_1 \cup \Gamma_2)$
$M_1 \bullet M_2$ *est la matrice d'adjacence du graphe* $G'=(X, \Gamma_2 \bullet \Gamma_1)$.

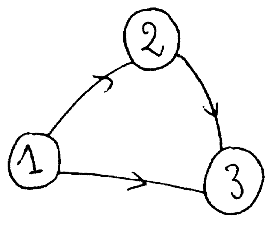
5°) Représentations graphiques de $G=(X, \Gamma_1 \cup \Gamma_2)$ *et de* $G'=(X, \Gamma_2 \bullet \Gamma_1)$.





TD2 III.4

*** Exercice 1.**



Soit le graphe $G=(X,\Gamma)$ représenté ci-contre.

1°)Sans utiliser le calcul matriciel, écrire M puis $M^2=M.M$.
Définir Γ puis Γ^2.
Vérifier que M^2 est la matrice du graphe $G'=(X, \Gamma^2)$.

2°) Ecrire la fermeture transitive et sa matrice associée.

Corrigé :
1°) A partir de la représentation géométrique du graphe on obtient :
- La matrice $M^2=(c_{ij})$ définie par :
$c_{ij}=1$ s'il existe (au moins) un chemin de longueur 2 de i à j.
$c_{ij}=0$ sinon.

- *L'application Γ^2 définie par $\Gamma^2(x)$:extrémités des chemins de longueur 2 pour tout sommet x.*

$$M=\begin{pmatrix}0&1&1\\0&0&1\\0&0&0\end{pmatrix};\quad M^2=\begin{pmatrix}0&0&1\\0&0&0\\0&0&0\end{pmatrix}$$

x	$\Gamma(x)$
1	{2,3}
2	{3}
3	∅

x	$\Gamma^2(x)$
1	{3}
2	∅
3	∅

On vérifie que M^2 est la matrice d'adjacence de $G=(X,\Gamma^2)$

Cet exemple illustre la propriété suivante :

M^k est la matrice de $G=(X,\Gamma^k)$, Γ^k étant définie de façon récurrente par :
$\forall x \in X, \Gamma^k(x) = \Gamma(\Gamma^{k-1}(x))$ *extrémités des chemins de longueur k à partir du sommet x.*

2°) La fermeture transitive de G peut être définie par l'application Γ' telle que :
$\Gamma'(x)= \Gamma(x) U \Gamma^2(x)$ pour tout sommet x (ensemble des sommets accessibles à partir de x ou «descendants» de x).

x	$\Gamma(x)U\Gamma^2(x)$
1	{2,3}
2	{3}
3	∅

Le graphe $G'=(X,\Gamma')$ est donc composé d'arcs reliant tout sommet x de G à l'ensemble de ses descendants. Il est transitif. G' est appelé fermeture transitive de G.
Sa matrice d'adjacence (matrice de la fermeture transitive) est obtenue à partir du tableau précédent.

La matrice de la fermeture transitive : $\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

Notons que les liaisons entre graphes et calcul booléen (vu au TD 1) nous permettent décrire $M'=M+M^2$ puis que $\Gamma'(x)=\Gamma(x) U \Gamma^2(x)$.

On constate que G=G', résultat prévisible, le graphe initial étant transitif.

* **Exercice 2** . Soit le graphe G=(X,Γ), Γ étant définie par le tableau suivant :

x	Γ(x)
1	{2,3}
2	{4}
3	{4}
4	$\varnothing$

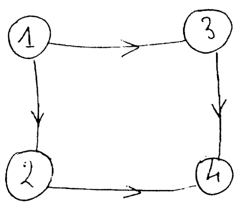
1°) Représentation géométrique du graphe.
Ecrire la matrice d'adjacence M du graphe G.

2°) Déterminer matriciellement le nombre de chemins de longueur 2. Entre tout couple de sommets du graphe.

3°) Définir Γ^2 à partir du tableau initial, en donner la signification.
Ecrire la matrice d'adjacence du graphe G=(X, Γ^2).
Calculer M^2, conclure.

4°) Calculer M^3 · Définir la fermeture transitive G' du graphe G et calculer sa matrice d'adjacence. Interprétation.

Corrigé:
1°)



$$M = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

2°) $M^2 = \begin{pmatrix} 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ *Il existe 2 chemins de longueur 2 entre les sommets 1 et 4 et aucun pour tout autre couple de sommets, ce que dénote la représentation graphique: chemins (1,2,4) et (1,3,4).*

3°) $\Gamma^2(x)=\Gamma(\Gamma(x))$
Par exemple
$\Gamma^2(1)=\Gamma(\Gamma(1))=\Gamma(\{2,3\})=\{4\}$.

$\Gamma^2(x)$ est l'ensemble des extrémités des chemins de longueur 2 ayant x pour origine.

x	*$\Gamma^2(x)$*
1	*{4}*
2	$\varnothing$
3	$\varnothing$
4	$\varnothing$

Matrice d'adjacence de $G=(X,\Gamma^2)$: $\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

M^2 est la matrice précédente, on peut conclure que M^2 est la matrice d'adjacence de $G=(X,\Gamma^2)$

.

4°) $M^3=0$. Il n'existe donc aucun chemin de longueur 3.
Fermeture transitive: $G'=(X,\Gamma')$ Γ' étant définie par $\Gamma'(x)=\Gamma(x) U \Gamma^2(x)$ (ensemble des sommets accesibles à partir de x).

La matrice d'adjacence de G' s'écrit:

$$M' = M + M^2 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Interprétation:
La première ligne de la matrice M' fournit les sommets accessibles à partir du sommet 1: 2,3,4.
En considérant les autres lignes de M': 4 est accessible à partir des sommets 2 et 3, aucun sommet n'étant accessible à partir de 4.

Thème d'étude

** III.5: Fermeture transitive par l'Algorithme de Warshall (1962)[12]

L'algorithme suivant convertit la matrice d'adjacence A d'un graphe simple orienté en la matrice d'adjacence de la fermeture transitive. Il s'appuie sur la définition classique de la transitivité.

Algorithme: Fermeture transitive (Warshall)

Donnée: Un graphe simple orienté et sa matrice d'adjacence A.
Résultat: La matrice de la fermeture transitive A.

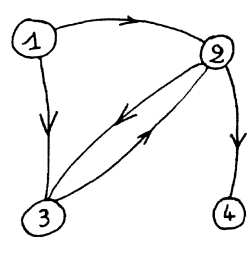
```
Pour j variant de 1 à n.
    Pour i variant de 1 à n.
        Si A[i,j]=1 alors
            Pour k variant de 1 à n
                Si A[j,k]=1 alors A[i,k]=1.
```

Validité et complexité.
Le plus grand problème à résoudre pour cet algorithme est le suivant:
Une seule passe est-elle suffisante? Les modifications apportées à la matrice A au cours d'une itération sont-elles prises en compte au cours des itérations suivantes?

La réponse est oui, en effet:
Si une modification de la matrice est effectuée au cours de l'itération j, elle ne concerne que la ligne i de la matrice et n'ajoute donc au graphe que des arcs d'origine i (on fait en réalité un OU booléen entre la ligne i et la ligne j). Cette modification n'est d'aucune utilité pour les itérations i+1,i+2,...De plus elle sera prise en compte au cours des itérations j+1,j+2,...

Complexité de l'algorithme $O(n^2+nm)$

Exempl : (détaillé dans l'exercie III.6)

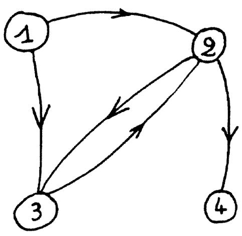


La matrice A dans différents états:
Initialement, après l'itération j=2 et après l'itération j=3.
Les modifications apportées à chaque itération sont présentées en caractère gras.

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} ; \text{ Après } j=2 \begin{pmatrix} 0 & 1 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 1 & 1 \\ 0 & 1 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & 0 \end{pmatrix} ; \text{ Après } j=3 \begin{pmatrix} 0 & \mathbf{1} & 1 & 1 \\ 0 & \mathbf{1} & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

[12]R.Sedgewick, *Algorithms* second edition, p 475, 476, Addison-Wesley

** **Exercice III.6** On considère le graphe simple orienté suivant :



1°) Donner la matrice d'adjacence A du graphe.
2°) Pour les valeurs de i variant de 1 à 4 expliciter l'algorithme de Warshall pour les itérations j=2 puis j=3.
Ecrire les matrices successives obtenues.
3°) Vérifier que la matrice finale obtenue représente bien la fermeture transitive du graphe.

Corrigé :1°)

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix};$$

2°) l'itération j=1 n'apporte pas de modification à la matrice.

i	*j=1*	*j=2*	*j=3*
i=1		$a_{12}=1$;$a_{24}=1$ *d'où* $a_{14}=1$	
i=2			$a_{23}=1$;$a_{32}=1$ *d'où* $a_{22}=1$
i=3		$a_{32}=1$;$a_{23}=1$ *d'où* $a_{33}=1$ $a_{32}=1$;$a_{24}=1$ *d'où* $a_{34}=1$	
i=4			

3°)

$$;\text{Après } j = 2 \begin{pmatrix} 0 & 1 & 1 & \mathbf{1} \\ 0 & 0 & 1 & 1 \\ 0 & 1 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & 0 \end{pmatrix}; \ \text{Après } j = 3 \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & \mathbf{1} & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} = A'$$

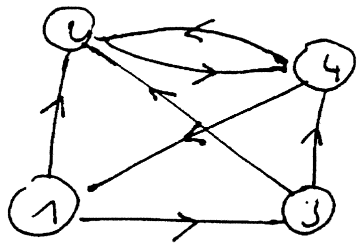
3°) La matrice finale représente bien la fermeture transitive du graphe. En effet, la représentation géométrique montre que les sommets accessibles à partir du sommet 1 sont les sommets 2,3,4 ; de même pour les sommets 2 et 3, aucun sommet n'étant accessible à partir de 4.

Une vérification matricielle peut –être opérée

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; A^2 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \ A^3 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

On vérifie bien que $A+A^2+A^3=A'$.

** **Exercice III.7** Mêmes questions avec :



Corrigé :

i	j=1	j=2	j=3	j=4
i=1		a_{12}=1,a_{24}=1 donc a_{14}=1		a_{14}=1,a_{41}=1 donc a_{11}=1
i=2				a_{24}=1,a_{41}=1 donc a_{21}=1 a_{24}=1,a_{42}=1 donc a_{22}=1 a_{24}=1,a_{43}=1 donc a_{23}=1
i=3		a_{32}=1,a_{24}=1 donc a_{34}=1		a_{34}=1,a_{41}=1 donc a_{31}=1 a_{34}=1,a_{43}=1 donc a_{33}=1
i=4	a_{41}=1,a_{13}=1 donc a_{43}=1	a_{42}=1,a_{24}=1 donc a_{44}=1		

Après j=1 Après j=2 Après j=3 Après j=4.

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}; \quad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}; \quad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}; \quad \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} = A'$$

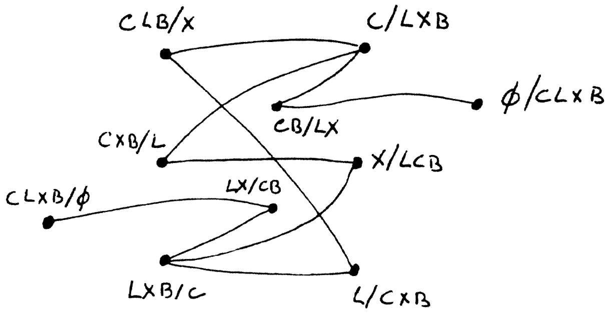
3°) On vérifie aisément que tous les sommets sont accessibles à partir d'un sommet donné. Matriciellement on vérifie également que : $A+A^2+A^3=A'$

IV Le problème du plus court chemin dans un graphe simple orienté valué.

- Exercices
- Pour tout couple de sommets : Algorithme de Floyd.
- A partir d'un sommet initial
  pour des valuations positives.
  (Algorithme de Moore-Dijsktra)
- Algorithme de Dantzig

Exercices

Exercice IV.1: Une solution en termes de graphe au célèbre problème du loup, de la chèvre et du chou.
Au bord d'une rivière se trouvent un loup, une chèvre, un chou et un batelier. Comment les faire traverser sachant qu'un seul passager peut être embarqué à la fois, tout en ménageant la chèvre et le chou (en évitant de laisser seuls le loup et la chèvre ainsi que la chèvre et le chou)?

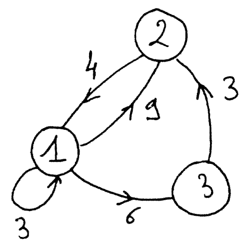


On peut représenter le problème par un graphe simple non orienté dont les sommets sont les différents états possibles sur les rives par exemple $CXLB/\varnothing$ *pour la situation initiale et LX/CB pour le loup et le chou sont seuls sur la rive initiale tandis que la chèvre et le batelier sont sur l'autre rive.*
Les états CX/LB, LC/ BX, LB/CX et BX/LC sont exclus du graphe.
Les arêtes de ce graphe sont les transitions possibles d'un état à un autre.
Le problème revient alors à rechercher un (plus court) chemin du sommet $CLXB/\varnothing$ *au sommet* $\varnothing$ / CXLB .

Exercice IV.2 Comment traiter le problème du plus court chemin dans le cas d'un multigraphe?
Solution: Extraire du multigraphe un graphe simple en ne conservant que l'arc de moindre coût.

Exercice IV.3: Comment traiter le problème du plus **long** chemin?
Solution: Ce problème n'a de solution que dans un graphe sans circuit de longueur positive. Il suffit de prendre l'opposé de chaque valeur d'arc et de chercher le plus court chemin.

**** Exercice IV.4 : Algorithme de Floyd.**



On considère le graphe simple orienté valué représenté ci-dessus.
1°) Ecrire la matrice initiale $A=A^{(0)}$ .de l'algorithme de Floyd.
2°) On note $A^{(k)}=(a_{ij}^{(k)})$. Montrer que dans un graphe à valuations positives:

a) Les termes diagonaux de toute matrice sont nuls : $\forall k,\ a_{ii}^{(k)} = 0$.

b) Si $i = k$ ou $j = k$, $a_{ij}^{(k)} = a_{ij}^{(k-1)}$ Les termes correspondants a_{kj} et a_{ik} sont inchangés d'une matrice d'ordre k-1 à la matrice d'ordre k.

3°) A l'aide de ces remarques déterminer $A^{(1)}, A^{(2)}, A^{(3)}$

Corrigé.

1°) $A = A^0 = \begin{pmatrix} 0 & 9 & 6 \\ 4 & 0 & \infty \\ \infty & 3 & 0 \end{pmatrix}$

2°) a) $a_{ii}^{(0)} = 0; a_{ij}^{(k)} = \min(a_{ij}^{(k-1)}; a_{ik}^{(k-1)} + a_{kj}^{(k-1)})$ *(Formule de base de l'algorithme de Floyd).*

Les arcs ayant une valuation positive, il résulte que :

Si $a_{ii}^{(k-1)} = 0$ alors $a_{ii}^{(k)} = \min(a_{ii}^{(k-1)}; a_{ik}^{(k-1)} + a_{ki}^{(k-1)}) = 0$

On a ainsi montré par récurrence que : $\forall k,\ a_{ii}^{(k)} = 0$.

b) Si j=k alors $a_{ij}^{(k)} = \min(a_{ij}^{(k-1)}; a_{ij}^{(k-1)} + a_{jj}^{(k-1)}) = \min(a_{ij}^{(k-1)}; a_{ij}^{(k-1)} + 0) = a_{ij}^{(k-1)}$

De même si i=k.

3°) $A = A^0 = \begin{pmatrix} 0 & 9 & 6 \\ 4 & 0 & \infty \\ \infty & 3 & 0 \end{pmatrix} \quad A^{(1)} = \begin{pmatrix} 0 & 9 & 6 \\ 4 & 0 & 10 \\ \infty & 3 & 0 \end{pmatrix}$

En effet puisque k=1, les termes $a_{12}, a_{13}, a_{21}, a_{31}$ sont inchangés et

$a_{23}^{(1)} = \min(a_{23}^{(0)}, a_{21}^{(0)} + a_{13}^{(0)}) = \min(\infty, 4+6) = 10$

$a_{32}^{(1)} = \min(a_{32}^{(0)}, a_{31}^{(0)} + a_{12}^{(0)}) = \min(3, \infty + 9) = 3$

$$A^{(1)} = \begin{pmatrix} 0 & 9 & 6 \\ 4 & 0 & 10 \\ \infty & 3 & 0 \end{pmatrix} \quad A^{(2)} = \begin{pmatrix} 0 & 9 & 6 \\ 4 & 0 & 10 \\ 7 & 3 & 0 \end{pmatrix}$$

k=2. Les termes a_{21},a_{22},a_{23},a_{12},a_{32} sont inchangés.

$$a_{13}^{(2)} = \min(a_{13}^{(1)}, a_{12}^{(1)} + a_{23}^{(1)}) = \min(6, 9+10) = 6$$

$$a_{31}^{(2)} = \min(a_{31}^{(1)}, a_{32}^{(1)} + a_{21}^{(1)}) = \min(\infty, 3+4) = 7$$

$$A^{(2)} = \begin{pmatrix} 0 & 9 & 6 \\ 4 & 0 & 10 \\ 7 & 3 & 0 \end{pmatrix} \quad A^{(3)} = \begin{pmatrix} 0 & 9 & 6 \\ 4 & 0 & 10 \\ 7 & 3 & 0 \end{pmatrix}$$

k=3. Les termes a_{31},a_{32},a_{33},a_{13},a_{23} sont inchangés.

$$a_{12}^{(3)} = \min(9, 6+3) = 9$$

$$a_{21}^{(3)} = \min(4, 10+7) = 4 .$$

***Thème d'Etude IV.5: Algorithme de Moore Dijsktra.(1957,1959)

Cet algorithme s'applique dans un graphe dont toutes les valeurs des arcs sont **positives,** il calcule le plus court chemin du sommet 1 à tous les autres sommets.
L'idée de base est de construire un vecteur π ayant n coordonnées tel que $\pi(i)$ représente la longueur du plus court chemin de 1 à i.
Le vecteur π est initialisé comme suit:

$$\begin{cases} \pi(1) = 0 \\ \text{pour chaque successeur i de 1; } \pi(i) = \text{la valeur de l'arc } (1,i). \\ \text{pour tout autre sommet i , } \pi(i) = \infty. \end{cases}$$

On considère également un ensemble S de sommets qui initialement ne contient que le sommet 1. Puis on ajoute pas à pas des sommets dans l'ensemble S jusqu'à ce qu'il contienne tous les sommets en faisant en sorte que les coordonnées du vecteur π soient valides pour les sommets de S.

Algorithme Plus court chemin (Moore-Dijsktra).

Donnée; Un graphe orienté valué à valeurs positives.
Résultat: Un vecteur π des plus courts chemins.

1ère Etape: Initialisations

Initialiser le vecteur π comme indiqué plus haut.
S={1}.; $\overline{S} = \{2,...n\}$.

2ème Etape; Itérations.

Tant que l'ensemble $\overline{S}$ n'est pas vide.
  Choisir i dans $\overline{S}$ tel que $\pi(i)$ est minimum.
  Retirer i de $\overline{S}$ et l'ajouter à S.
  Pour chaque successeur j de i dans $\overline{S}$
    $\pi(j)= \min(\pi(j), \pi(i)+$ valeur de l'arc (i,j)).

Validité et complexité de l'algorithme: A chaque itération de la boucle de la 2ème étape, on retire un sommet de $\overline{S}$, cette boucle est exécutée exactement n-1 fois, on est donc assuré de la terminaison de l'algorithme. Dans $\pi(i)$ on a la distance de 1 à i en passant par les sommets de S, l'arrivée de i dans S risque de modifier cette distance pour les successeurs de i, d'où la mise à jour de π en fin d'itération. De plus on a la propriété suivante qui est conservée à chaque itération:
Pour chaque sommet i de S, $\pi(i)$ est le plus court chemin de 1 à i dans le graphe restreint aux sommets de S.
Comme cette propriété est vraie à l'initialisation (S={1} et $\pi(1)$=0) elle demeure vraie à la fin de l'algorithme lorsque S contient tous les sommets du graphe.

Complexité:
Initialisation du vecteur π: O(n)

Initialisation de S et de $\overline{S}$: O(n), une partie de {1,...,n} peut facilement être décrite par un vecteur caractéristique de dimension n, les opérations d'ajout et de suppression d'un élément sont alors de complexité constante.

Choix de i: C'est le point faible de l'algorithme, il faut pouvoir trouver rapidement le minimum des $\pi(i)$ pour i dans $\overline{S}$. Des techniques avancées de gestion de files de priorité(ou tas) trop complexes pour être exposées ici permettent de le faire en O(logn)[13], mais cela suppose l'initialisation du tas O(nlogn) et son maintien O(logn) à chaque insertion ou suppression.

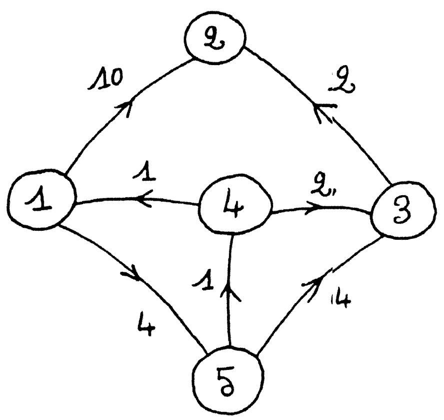
Transfert de i de $\overline{S}$ vers S: O(1).

La dernière boucle est exécutée une fois pour chaque arc tout au long de l'algorithme, pour chaque $\pi(j)$ modifié il faut payer d'opérations de gestion du tas pour une complexité de O(mlogn)

Ce qui conduit à une complexité de l'algorithme de O(nlogn+mlogn).

En adaptant l'algorithme de Moore Dijsktra pour qu'il fonctionne à partir de n'importe quel sommet et en l'appliquant à chaque sommet on obtient un résultat analoque à celui de l'algorithme de Floyd avec une complexité de O(n(n+m)logn).

Exemple:



Initialisation
$S=\{1\};\ \overline{S}=\{2,3,4,5\};\ \pi=(0,10,\infty,\infty,4)$.
1ère itération: i=5
$S=\{1,5\};\ \overline{S}=\{2,3,4\};\ \pi=(0,10,8,5,4)$/
2ème itération: i=4
$S=\{1,5,4\};\ \overline{S}=\{2,3\};\ \pi=(0,10,7,5,4)$.
3ème itération i=3
$S=\{1,5,4,3\};\ \overline{S}=\{2\};\ \pi=(0,9,7,5,4)$.
4ème itération i=2.
$S=\{1,5,4,3,2\};\ \overline{S}=\varnothing;\ \pi=(0,9,7,5,4)$.

[13] R.Sedgewick, *Algorithms* second edition, Addison-Wesley

***Thème d'étude IV.6: Algorithme de Dantzig (1966).

L'algorithme de Dantzig est un algorithme matriciel qui calcule le plus court chemin de n'importe quel sommet vers n'importe quel autre sommet et qui s'arrête dès qu'un circuit de longueur négative est trouvé.
L'algorithme est itératif, à la kème itération il fournit une matrice $A^{(k)}$ de taille kxk des plus courts chemins valides **dans le graphe restreint aux k premiers sommets**.
Il est à noter que la matrice $A^{(k)}$ utilisée ici est différente de celle utilisée dans l'algorithme de Floyd : le coefficient $a^{(k)}_{ij}$ donne le plus court chemin de i vers j pour i et j inférieurs à k. Ce coefficient n'a pas de signification pour i ou j supérieur à k.

La solution est donnée par la matrice $A^{(n)}$.

$$A^{(k+1)} = \begin{pmatrix} \begin{pmatrix} & & \\ & A^{(k)} & \\ & & \end{pmatrix} & \begin{matrix} a^{(k+1)}_{1,k+1} \\ . \\ . \\ a^{(k+1)}_{i,k+1} \\ . \end{matrix} \\ a^{(k+1)}_{k+1,1} \;\; .. \;\; a^{(k+1)}_{k+1,j} & a^{(k+1)}_{k+1,k+1} \end{pmatrix}$$

Le passage de la matrice $A^{(k)}$ à la matrice $A^{(k+1)}$ se fait en 4 étapes:
1°) Le calcul des $a^{(k+1)}_{i,k+1}$ par la formule

$$a^{(k+1)}_{i,k+1} = \min_{1\le j\le k}(a^{(k)}_{i,j} + \text{valeur de l'arc } (j,k+1))$$

Le plus court chemin de i à k+1 passe nécessairement par un sommet j inférieur à k. On calcule donc $a^{(k+1)}_{i,k+1}$ en prenant le plus court des chemins de i à k+1 en passant par un quelconque sommet j inférieur à k.
2°) Le calcul des $a^{(k+1)}_{k+1,j}$ par la formule

$$a^{(k+1)}_{k+1,j} = \min_{1\le i\le k}(\text{valeur de l'arc } (k+1,i) + a^{(k)}_{i,j})$$

3°) **Les coefficients** $a^{(k)}_{i,j}$ **ont besoin d'être recalculés**: il est possible que le chemin de i à j soit plus court en passant par k+1

$$a^{(k+1)}_{i,j} = \min(a^{(k)}_{i,j} \,,\; a^{(k+1)}_{i,k+1} + a^{(k+1)}_{k+1,j})$$

4°) En conditions normales le coefficient $a^{(k+1)}_{k+1,k+1}$ vaut 0, la formule suivante permet de détecter un circuit de longueur négative.

$$a^{(k+1)}_{k+1,k+1} = \min_{1\le i\le k}(\,0 \,,\; a^{(k+1)}_{k+1,i} + a^{(k+1)}_{i,k+1})$$

En effet si $a^{(k+1)}_{k+1,k+1}$ est négatif après le calcul précédent c'est que le sommet correspondant se trouve sur un circuit de longueur négative.

Comme pour l'algorithme de Floyd on n'a pas besoin de conserver en mémoire les matrices intermédiaires, on peut surcharger la même matrice au fur et à mesure des itérations, l'ordre des calculs est alors important.

Algorithme Plus court chemin (Dantzig)

Donnée Un graphe orienté simple valué.

Résultat: La matrice A des plus courts chemins ou le message «Circuit détecté.»

Initialisation: $a_{1,1} = 0$.

Itérations: Pour k variant de 2 à n

$$a_{i,k+1} = \min_{1\le j\le k}(a_{i,j} + \text{valeur de l'arc } (j,k+1))$$

$$a_{k+1,j} = \min_{1\le i\le k}(\text{valeur de l'arc } (k+1,i) + a_{i,j})$$

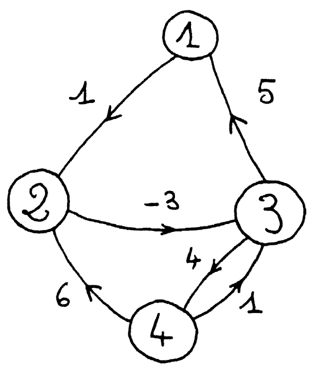
$$a_{i,j} = \min(a_{i,j}\ ,\ a_{i,k+1} + a_{k+1,j})$$

$$a_{k+1,k+1} = \min_{1\le i\le k}(\ 0\ ,\ a_{k+1,i} + a_{i,k+1})$$

Si $a_{k+1,k+1} < 0$ Arrêter «Circuit détecté»

Complexité :$O(n^3)$

Exemple:



La matrice des valeurs des arcs:

$$\begin{pmatrix} 0 & 1 & \infty & \infty \\ \infty & 0 & -3 & \infty \\ 5 & \infty & 0 & 1 \\ \infty & 6 & 4 & 0 \end{pmatrix}$$

Les matrices $A^{(k)}$ successives.

$$A^{(1)} = \begin{pmatrix} 0 & X & X & X \\ X & X & X & X \\ X & X & X & X \\ X & X & X & X \end{pmatrix};\ A^{(2)} = \begin{pmatrix} 0 & 1 & X & X \\ \infty & 0 & X & X \\ X & X & X & X \\ X & X & X & X \end{pmatrix}$$

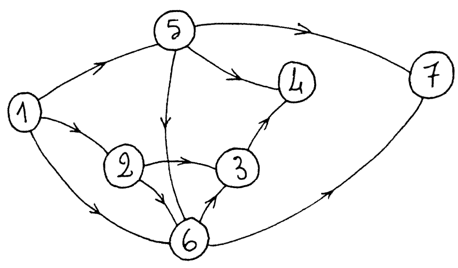
$$A^{(3)} = \begin{pmatrix} 0 & 1 & -2 & X \\ 2 & 0 & -3 & X \\ 5 & 6 & 0 & X \\ X & X & X & X \end{pmatrix};\ A^{(1)} = \begin{pmatrix} 0 & 1 & -2 & 2 \\ 2 & 0 & -3 & 1 \\ 5 & 6 & 0 & 1 \\ 8 & 6 & 3 & 0 \end{pmatrix}$$

V Ordonnancement par niveaux d'un graphe simple orienté sans circuit.

- Exercices.
- Une approche matricielle de l'ordonnancement par niveaux des sommets.

Exercices.

* **Exercice V.1** On considère le graphe G suivant



1°) Déterminer les niveaux (rangs)des sommets du graphe en utilisant la définition du niveau d'un sommet à l'aide de la représentation géométrique.
2°)Ordonnancer le graphe par niveaux par une méthode algorithmique que l'on précisera, le représenter.

Corrigé :

1°) *Remarquons que le graphe est un graphe fini simple orienté sans circuit, et qu'il n'y a qu'un seul sommet sans prédécesseur, le sommet 1.*
Le rang (ou niveau), d'un sommet i est la cardinalité maximale d'un chemin de 1 à i ou encore la longueur du plus long chemin ayant pour extrémité i.
La représentation géométrique du graphe nous permet d'écrire :
1 a pour niveau 0
2 et 5 ont pour niveau 1
6 a pour niveau 2
3 et 7 ont pour niveau 3
4 a pour niveau 4

2°) *Rappelons l'algorithme de calcul du rang d'un sommet valable dans un graphe simple orienté sans circuit ayant un unique sommet sans prédécesseur.*

Algorithme Rang des sommets.
***Données**: Un graphe simple orienté sans circuit avec un unique sommet sans prédécesseur x_0*
***Résultat**: . Le rang de chaque sommet*

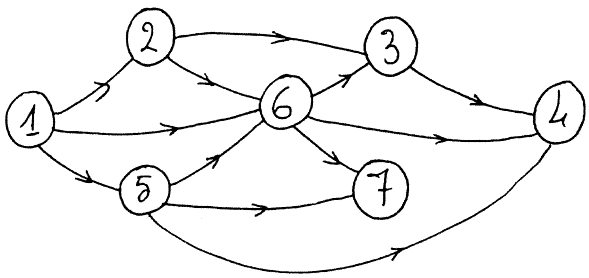
Rang courant=0
Tant que le graphe n'est pas vide
Tous les sommets sans prédécesseur ont le rang courant.
Retirer du graphe tous les sommets sans prédécesseur.
Incrémenter le rang courant.

On peut appliquer cette procédure à partir du tableau des prédécesseurs :

x	$\Gamma^{-1}(x)$
1	$\varnothing$
2	{1}
3	{2,6}
4	{3,5}
5	{1}
6	{1,2,5}
7	{5,6}

On réalise alors une partition de l'ensemble des sommets en niveau.
N_0 est l'ensemble des sommets de niveau 0,
seul 1 n'a pas de prédécesseur donc $N_0=\{1\}$
On retire du graphe tous les sommets sans prédécesseur : On élimine donc les sommets de niveau 0 dans la colonne des prédécesseurs $\Gamma^{-1}(x)$ (On barre 1 dans l'exemple donné)
Les sommets 2 et 5 n'ayant plus de prédécesseur sont de niveau 1 : $N_1=\{2,5\}$.
On réitère la procédure et ainsi : $N_0=\{1\}$, $N_1=\{2,5\}$. ;$N_2=\{6\}$;$N_3=\{3,7\}$;$N_4=\{4\}$.

Représentation du graphe ordonnancé par niveaux.



***Exercice V.2** Comment détecter un circuit dans un graphe en utilisant la propriété suivante?
Dans un graphe simple orienté sans circuit, il existe toujours un sommet sans prédécesseur.

Solution

> ***Algorithme** Détection d'un circuit;*
>   ***Donnée:** Un graphe simple orienté.*
>   ***Résultat:** Le message «Circuit détecté» ou «Le graphe est sans circuit»*
>
>   *Tant qu'il reste un sommet sans prédécesseur*
>   *Retirer ce sommet du graphe.*
>   *S'il reste des sommets dans le graphe*
>   *alors «Circuit détecté»*
>   *Sinon «Le graphe est sans circuit».*

Chaque nouvelle itération de la boucle retire au moins un sommet, la terminaison est donc assurée.
Si un graphe est sans circuit il sera vidé complètement par l'algorithme.
Si un graphe possède un circuit, tout sommet de ce circuit a un prédécesseur et donc aucun ne peut être retiré du graphe.
Complexité O(n+m) si le graphe est donné par listes d'adjacence.

***Exercice V.3**: Démontrer qu'un graphe simple orienté sans circuit avec un unique sommet sans prédécesseur est nécessairement connexe.
Solution : Si le graphe n'était pas connexe, il aurait au moins 2 composantes connexes elles aussi sans circuit. Chaque composante connexe aurait au moins un sommet sans prédécesseur, et le graphe aurait plus d'un sommet sans prédécesseur.

Une approche matricielle de l'ordonnancement d'un graphe par niveaux.

Il est facile de reconnaître un sommet sans prédécesseur sur une matrice d'adjacence : le vecteur colonne correspondant à ce sommet est nul. L'exercice suivant s'appuie sur cette idée.

*Exercice V.4 Soit le graphe G=(X, Γ) défini par X={1,2,3,4,5} et par le tableau des successeurs suivant :

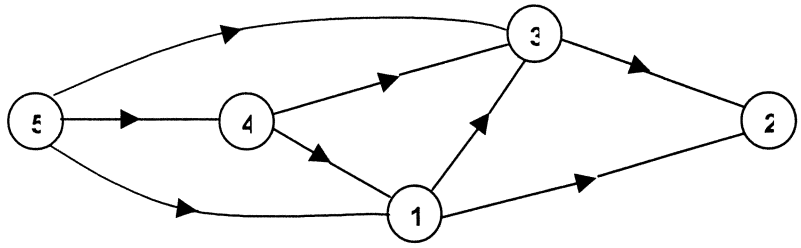
x	Γ(x)
1	{2,3}
2	∅
3	{2}
4	{1,3}
5	{1,3,4}

1°) Ecrire la matrice d'adjacence M du graphe G.
2°) Calculer les puissances de M. En déduire le niveau des sommets, justifiez.
3°) Retrouver les résultats précédents en utilisant l'algorithme de calcul du rang d'un sommet à partir du tableau
4°)Représenter le graphe ordonnancé par niveaux.

Corrigé :

$$M=\begin{pmatrix}0&1&1&0&0\\0&0&0&0&0\\0&1&0&0&0\\1&0&1&0&0\\1&0&1&1&0\end{pmatrix};M^2=\begin{pmatrix}0&1&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&2&1&0&0\\1&2&2&0&0\end{pmatrix};M^3=\begin{pmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&1&0&0&0\\0&3&1&0&0\end{pmatrix};M^4=\begin{pmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&1&0&0&0\end{pmatrix};M^5=0$$

Notons que M^5 étant nulle, le graphe ne comporte pas de circuit. On peut donc l'ordonnancer par niveaux.
Le $5^{ème}$ vecteur colonne de la matrice M est nul, le sommet 5 est donc sans prédécesseur, il est de niveau 0 et c'est le seul : $N_0=\{5\}$.
Le $4^{ème}$ vecteur colonne de M² est nul, il n'existe donc aucun chemin de longueur 2 ayant 4 pour extrémité, comme d'autre part le $4^{ème}$ vecteur colonne de M n'est pas nul, il existe un chemin de longueur 1 ayant 4 pour extrémité. Par définition du niveau d'un sommet, 4 est de niveau 1. $N_1=\{4\}$
De la même façon on obtient :$N_2=\{1\}$;$N_3=\{3\}$;$N_4=\{2\}$.

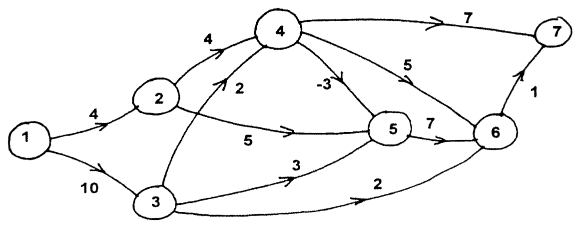


VI Détermination des chemins de valeur minimale et maximale à partir d'un sommet donné, le graphe étant ordonnancé par niveaux.

- Algorithme de Bellman.
- Méthode Electre.

Algorithme de Bellman.

*** Exercice VI.1** On considère le graphe G ci-dessous (graphe simple orienté sans circuit ordonnancé par niveaux).



1°) On souhaite déterminer la valeur du plus court chemin du sommet 1 au sommet 7 à l'aide de l'algorithme de Bellman donné ci-dessous.:

Expliciter l'algorithme pour les quatre premières valeurs de i : i=1,3,2,4.

Algorithme Plus court chemin (Bellman)

Donnée: Un graphe simple orienté sans circuit avec une unique racine x_0.

Résultat: Un vecteur π des plus courts chemins à partir de x_0.

$\pi(x_0)=0$; $\pi(i)=\infty$ pour les autres sommets.
Pour chaque sommet i (dans l'ordre des rangs)
Pour chaque successeur j de i
$\pi(j)=\min(\pi(j),\pi(i)+$ valeur de l'arc $(i,j))$

2°) Même question en utilisant l'algorithme suivant :

Marque du sommet initial $\pi(1)=0$.
Pour chaque sommet j dans l'ordre des rangs
$\pi(j)= \min(\pi(i)+$valeur de l'arc $(i,j))$ pour chaque sommet i prédécesseur de j

Corrigé :

	$\pi(1)$	$\pi(2)$	$\pi(3)$	$\pi(4)$	$\pi(5)$	$\pi(6)$	$\pi(7)$
Init	*0*	∞	∞	∞	∞	∞	∞
I=1	*0*	*4*	*10*	∞	∞	∞	∞
I=3	*0*	*4*	*10*	*12*	*13*	*12*	∞
I=2	*0*	*4*	*10*	*8*	*9*	*12*	∞
I=4	*0*	*4*	*10*	*8*	*5*	*12*	*15*
I=5	*0*	*4*	*10*	*8*	*5*	*12*	*15*
Ii=6	*0*	*4*	*10*	*8*	*5*	*12*	*13*
I=7	*0*	*4*	*10*	*8*	*5*	*12*	*13*

Valeur du plus court chemin de 1 à 7 $\pi(7)=13$.

Pour i=1 les successeurs de 1 sont j=2 et j=3.

$\pi(2)=\min(\pi(2)\,;\pi(1)+$ valeur de l'arc $(1,2))=\min(\infty\,;0+4)=4.$

$\pi(3)=\min(\pi(3)\,;\pi(1)+$ valeur de l'arc $(1,3))=\min(\infty\,;0+10)=10.$

Pour i=3, les successeurs de 3 sont 4,5,et 6.

$\pi(4)=\min(\pi(4)\,;\pi(3)+$ valeur de l'arc $(3,4))=\min(\infty\,;10+2)=12.$

$\pi(5)=\min(\pi(5)\,;\pi(3)+$ valeur de l'arc $(3,5))=\min(\infty\,;10+3)=13.$

$\pi(6)=\min(\pi(6)\,;\pi(6)+$ valeur de l'arc $(3,6))=\min(\infty\,;10+2)=12.$

Pour i=2, les successeurs de 2 sont 4 et 5.

$\pi(4)=\min(\pi(4)\,;\pi(2)+$ valeur de l'arc $(2,4))=\min(12;4+4)=8.$

$\pi(5)=\min(\pi(5)\,;\pi(2)+$ valeur de l'arc $(2,5))=\min(13;4+5)=9.$

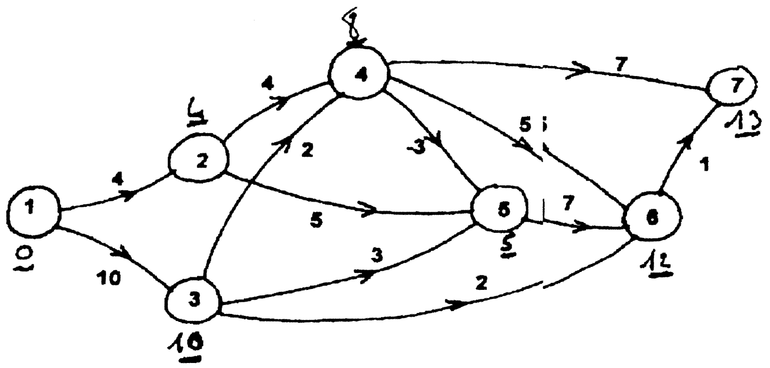
Pour i=4, les successeurs de 4 sont 5,6 et 7.

$\pi(5)=\min(\pi(5)\,;\pi(4)+$ valeur de l'arc $(4,5))=\min(9;8-3)=5.$

$\pi(6)=\min(\pi(6)\,;\pi(4)+$ valeur de l'arc $(4,6))=\min(12;8+5)=12.$

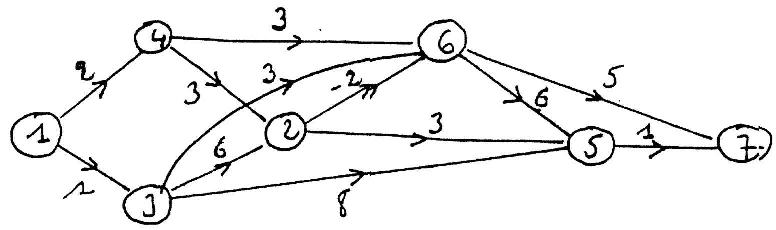
$\pi(7)=\min(\pi(7)\,;\pi(4)+$ valeur de l'arc $(4,7))=\min(\infty\,;8+7)=15.$

2°



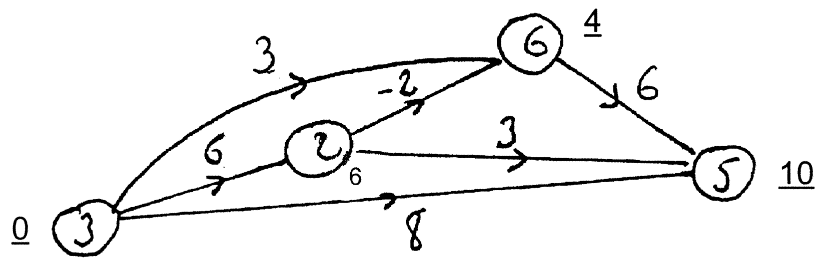
j	$\pi(j)$
1	*0*
2	$\pi(2)=\min(\pi(1)+\text{valeur de l'arc}(1,2))=4.$
3	$\pi(3)=\min(\pi(1)+\text{valeur de l'arc}(1,3))=10.$
4	$\pi(4)=\min(\pi(2)+\text{valeur de l'arc}(2,4);\pi(3)+\text{valeur de l'arc}(3,4))=\min(4+4,10+2)=8.$
5	$\pi(5)=\min(\pi(3)+3;\pi(2)+5;\pi(4)-3)=\min(10+3;4+5;8-3)=5.$
6	$\pi(6)=\min(\pi(3)+2;\pi(4)+5;\pi(5)+7)=\min(10+2;8+5;5+7)=12$
7	$\pi(7)=\min(\pi(4)+7;\pi(6)+1)=\min(8+7;12+1)=13.$

* **Exercice VI.2** On considère le graphe suivant ordonnancé par niveaux.



Déterminer le chemin de cardinalité maximale ayant pour origine 3 et pour extrémité 5.

Solution.
On considère le sous graphe obtenu en supprimant les sommets de niveau inférieur ou égal à celui de 3 et de niveau supérieur ou égal à celui de 5 (on conserve les sommets 3 et 5) ainsi que les arcs dont ils sont l'origine et l'extrémité.



Marque du sommet initial $\pi(3)=0$. Pour chaque sommet j dans l'ordre des rangs $\pi(j)=\max(\pi(i)+$valeur de l'arc $(i,j))$ pour chaque sommet i prédécesseur de j

La marque $\pi(5)=10$ donne la valeur maximale du chemin reliant 3 à 5; ce chemin est: (3,2,6,5).

*Exercice VI.3. Une société fait appel à des services de transport pour effectuer ses livraisons d'une ville A à une ville H.
Les liaisons possibles de ville à ville et les coûts sont traduits dans le tableau suivant :

	A	B	C	D	E	F	G	H
A			5	15				
B	15		20					
C				12			8	
D					25	20		30
E								12
F								10
G					25	17		
H								

1°) Présenter le problème sous forme d'un graphe. A cet effet
a) Donner la matrice d'adjacence du graphe et le tableau des prédécesseurs.
b) Déterminer le niveau des sommets.
c) En déduire la représentation géométrique du graphe ordonnancé par niveaux.

2°) Déterminer le coût minimal d'une livraison de A à H.

Corrigé:

1°)a)

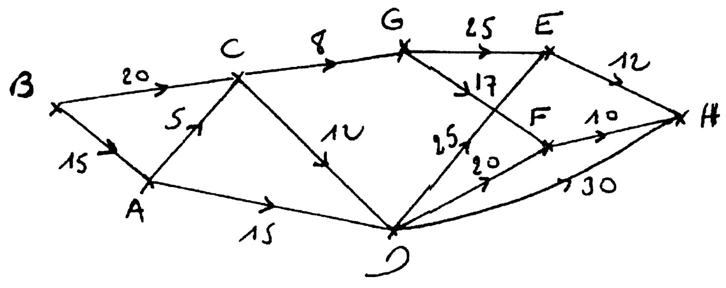
$$\begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

x	$\Gamma^{-1}(x)$
A	B
B	/
C	A,B
D	A,C
E	D,G
F	D,G
G	C
H	D,E;F

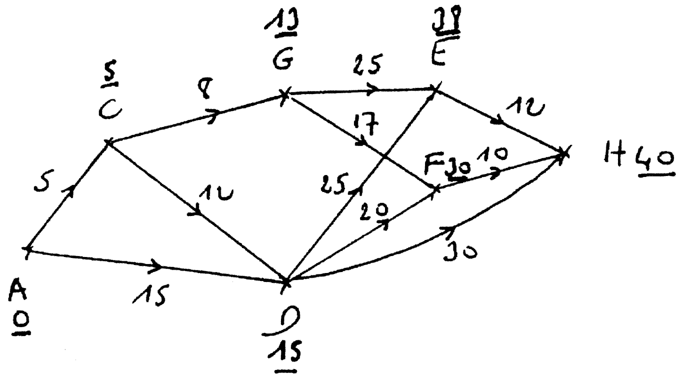
b) Niveaux des sommets obtenus à partir du tableau des prédécesseurs.

$N_0=\{B\}$; $N_1=\{A\}$ $N_2=\{C\}$; $N_3=\{D,G\}$; $N_4=\{E,F\}$; $N_5=\{H\}$.

c)Représentation du graphe ordonnancé par niveaux.



2°) La livraison devant s'effectuer de A à H, on est amené à supprimer le sommt B de niveau inférieur à celui de A et à considérer le sous graphe suivant:



En appliquant l'algorithme de Bellman simplifié, on détermine la marque de chaque sommet: coût le plus faible entre A et ce sommet.

$\pi(A)=0$

$\pi(C)=5$

$\pi(D)=min(12+5,15)=15.$

$\pi(G)=5+8=13.$

$\pi(E)=min(13+25;25+15)=38.$

$\pi(F)=min(15+20;13+17)=30.$

$\pi(H)=min(38+12;30+10;\ 15+30)=40.$

Le coût minimal d'une livraison de A à H est donc de 40, le chemin minimal étant A,C,G,F,H.

Méthode Electre

**** Thème d'Etude VI.2:. Application de la théorie des graphes à des problèmes de choix multicritères.**

La méthode Electre est une méthode d'aide à la décision par **EL**imination **E**t **C**hoix **T**raduisant la **RE**alité (B.Roy). Peut-être B. Roy souhaitait-il également faire allusion à la fille d'Agamemnon et de Clytemnestre, poussant son frère Oreste à cette décision difficile : tuer Egisthe et Clytemnestre, meurtriers de son père ?

Etude d'un exemple.

Une société de vente de matériel informatique a retenu sept ordinateurs de marques différentes, trois d'entre eux devant être sélectionnés en fonction de cinq critères :

1°) Le prix.
2°) La performance du processeur et des mémoires associées.
3°) La fiabilité (qualité des composants et de montages)
4°) La richesse des équipements multimédias.
5°) Le *design* et le confort de l'utilisateur (qualité d'écran et facilité d'usage)

Ces divers critères ayant la même importance relative compte tenu de la clientèle ciblée.

Une série de tests et d'informations ont permis d'associer à chaque ordinateur et pour chacun des 5 critères une valeur comprise entre 1 et 7. Le classement des ordinateurs étant fait par ordre décroissant de prix. Il résulte le tableau suivant :

Critères k / Ordinateurs xi	1	2	3	4	5
x_1	6	3	3.5	5	4
x_2	2	7	4	6	3.5
x_3	3	5	3	3.5	5
x_4	7	4	2	2	4
x_5	1	7	3.5	6	3
x_6	5	5	3.5	3.5	4
x_7	4	4	6	3.5	4

Pour permettre de comparer deux ordinateurs x_i et x_j, on définit la relation de «surclassement» suivante :

x_i surclasse x_j (On note x_i S x_j) si et seulement si x_i est préférable à x_j pour un critère et préférable ou équivalent à x_j pour au moins trois autres critères.

En considérant cette relation de surclassement, sélectionner trois ordinateurs parmi les sept retenus.

Corrigé :

Soit $K=\{1,2,3,4,5\}$ l'ensemble des critères et $f_k(x_i)$ l'évaluation de l'ordinateur xi pour le critère k de K.
La comparaison des ordinateurs 2 à 2 permet de définir les ensembles K^+, $K^=$ et K^- suivants :
$K^+=\{k|f_k(x_i)>f_k(x_j)\}$. Ensemble des critères favorables à x_i relativement à x_j.
$K^==\{k|f_k(x_i)=f_k(x_j)\}$. Ensemble des critères pour lesquels x_i et x_j ne se départagent pas
$K^-=k|f_k(x_i)<f_k(x_j)\}$. Ensemble des critères favorables à x_j relativement à x_i

Il résulte le tableau suivant :

x_j ----- x_i	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	$K^+=\{\}$ $K^==K$ $K^-=\{\}$	$K^+=\{1,5\}$ $K^==\{\}$ $K^-=\{2,3,4\}$	$K^+=\{1,3,4\}$ $K^==\{\}$ $K^-=\{2,5\}$	$K^+=\{3,4\}$ $K^==\{5\}$ $K^-=\{1,2\}$	$K^+=\{1,5\}$ $K^==\{3\}$ $K^-=\{2,4\}$	$K^+=\{1,4\}$ $K^==\{3,5\}$ $K^-=\{2\}$	$K^+=\{1,4\}$ $K^==\{5\}$ $K^-=\{2,3\}$
x_2	$K^+=\{2,3,4\}$ $K^==\{\}$ $K^-=\{1,5\}$	$K^+=\{\}$ $K^==K$ $K^-=\{\}$	$K^+=\{2,3,4\}$ $K^==\{\}$ $K^-=\{1,5\}$	$K^+=\{2,3,4\}$ $K^==\{\}$ $K^-=\{1,5\}$	$K^+=\{1,3,5\}$ $K^==\{2,4\}$ $K^-=\{\}$	$K^+=\{2,3,4\}$ $K^==\{\}$ $K^-=\{1,5\}$	$K^+=\{2,4\}$ $K^==\{\}$ $K^-=\{1,3,5\}$
x_3	$K^+=\{2,5\}$ $K^==\{\}$ $K^-=\{1,3,4\}$	$K^+=\{1,5\}$ $K^==\{\}$ $K^-=\{2,3,4\}$	$K^+=\{\}$ $K^==K$ $K^-=\{\}$	$K^+=\{2,3,4,5\}$ $K^==\{\}$ $K^-=\{1\}$	$K^+=\{1,5\}$ $K^==\{\}$ $K^-=\{2,3,4\}$	$K^+=\{5\}$ $K^==\{2,4\}$ $K^-=\{1,3\}$	$K^+=\{2,5\}$ $K^==\{4\}$ $K^-=\{1,3\}$
x_4	$K^+=\{1,2\}$ $K^==\{5\}$ $K^-=\{3,4\}$	$K^+=\{1,5\}$ $K^==\{\}$ $K^-=\{2,3,4\}$	$K^+=\{1\}$ $K^==\{\}$ $K^-=\{2,3,4,5\}$	$K^+=\{\}$ $K^==K$ $K^-=\{\}$	$K^+=\{1,5\}$ $K^==\{\}$ $K^-=\{2,3,4\}$	$K^+=\{1\}$ $K^==\{5\}$ $K^-=\{2,3,4\}$	$K^+=\{1\}$ $K^==\{2,5\}$ $K^-=\{3,4\}$
x_5	$K^+=\{2,4\}$ $K^==\{3\}$ $K^-=\{1,5\}$	$K^+=\{\}$ $K^==\{2,4\}$ $K^-=\{1,3,5\}$	$K^+=\{2,3,4\}$ $K^==\{\}$ $K^-=\{1,5\}$	$K^+=\{2,3,4\}$ $K^==\{\}$ $K^-=\{1,5\}$	$K^+=\{\}$ $K^==K$ $K^-=\{\}$	$K^+=\{2,4\}$ $K^==\{3\}$ $K^-=\{1,5\}$	$K^+=\{2,4\}$ $K^==\{\}$ $K^-=\{1,3,5\}$
x_6	$K^+=\{2\}$ $K^==\{3,5\}$ $K^-=\{14\}$	$K^+=\{1,5\}$ $K^==\{\}$ $K^-=\{2,3,4\}$	$K^+=\{1,3\}$ $K^==\{2,4\}$ $K^-=\{5\}$	$K^+=\{2,3,4\}$ $K^==\{5\}$ $K^-=\{1\}$	$K^+=\{1,5\}$ $K^==\{3\}$ $K^-=\{2,4\}$	$K^+=\{\}$ $K^==K$ $K^-=\{\}$	$K^+=\{1,2\}$ $K^==\{4,5\}$ $K^-=\{3\}$
x_7	$K^+=\{2,3\}$ $K^==\{5\}$ $K^-=\{1,4\}$	$K^+=\{1,3,5\}$ $K^==\{\}$ $K^-=\{2,5\}$	$K^+=\{1,3\}$ $K^==\{4\}$ $K^-=\{2,5\}$	$K^+=\{3,4\}$ $K^==\{2,5\}$ $K^-=\{1\}$	$K^+=\{1,3,5\}$ $K^==\{\}$ $K^-=\{2,4\}$	$K^+=\{3\}$ $K^==\{4,5\}$ $K^-=\{1,2\}$	$K^+=\{\}$ $K^==K$ $K^-=\{\}$

Du tableau précédent on déduit la matrice des valeurs c_{ij} de concordance (si K^+ est non vide , c_{ij} est le nombre de critères pour lesquels x_i est préférable ou équivalent à x_j),ainsi que le tableau de surclassement au seuil de concordance p=4.
(on a x_i S x_j si et seulement si $c_{ij} \geq 4$)

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1		2	3	3	3	4	3
x_2	3		3	3	5	3	2
x_3	2	2		4	2	3	3
x_4	3	2	1		2	2	3
x_5	3	2	3	3		3	2
x_6	3	2	4	4	3		4
x_7	3	3	3	4	32	3	

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1						S	
x_2					S		
x_3				S			
x_4							
x_5							
x_6			S	S			S
x_7				S			

On peut ainsi définir un graphe dont les sommets sont les ordinateurs x_i et les arcs définis par la relation de surclassement, la matrice d'adjacence du graphe s'écrivant : $a_{ij}=1$ si et seulement si x_i S x_j., 0 sinon

La fonction prédécesseur est obtenue à partir des colonnes de la matrice :

x_i	$\Gamma^{-1}(x)$
x_1	{}
x_2	{}
x_3	$\{x_6]$
x_4	$\{x_3,x_6,x_7\}$
x_5	$\{x_2\}$
x_6	$\{x_1\}$
x_7	$\{x_6\}$

Dans cet exemple le graphe est sans circuit ; on peut donc déterminer les niveaux :
$N_0=\{x_1,x_2\}$ $N_1=\{x_5,x_6\}$ $N_2=\{x_3,x_7\}$ $N_3=\{x_4\}$

La relation de surclassement nous amène donc au choix suivant :
les ordinateurs x_1, x_2 et l'un des deux ordinateurs x_5 ou x_6. Tout dépend en dernier ressort si on veut privilégier le prix ou la performance.

VII Problèmes d'ordonnancement.

- **Méthode P.E.R.T**
- **Vu à l'examen.**

(BTS Comptabilité 1986)

La méthode PERT

**** Exercice VII.1 Etude d'un exemple.**

L'entreprise X décide de lancer un produit nouveau sur le marché. Les services commerciaux ont déterminé l'ensemble des tâches nécessaires à cette action :a,b,c,d,e,f,g,h,i,j,k.

Les conditions d'antériorité liant ces tâches et les durées en jours de celles-ci sont rassemblées dans le tableau ci-dessous.

Tâches	Tâches antérieures	Durées
a	e	4
b	j,e	6
c	/	12
d	/	14
e	/	8
f	d	2
g	f,d	10
h	a,c,d	6
i	h,a,c,e,k,d,f	8
j	e	12
k	f,d	2

1) a) Déterminer le niveau des tâches
   b) Déterminer les tâches immédiatement antérieures à chaque tâche.

2) Tracer le graphe du projet par la méthode PERT en indiquant sur le graphe les dates de début au plus tôt et les dates de début au plus tard.
   En quel temps minimum le lancement du produit pourra t il être réalisé ?

3) Etablir un tableau des dates de début au plus tôt, des dates de début au plus tard des diverses tâches et des marges de flottement (marges totales).

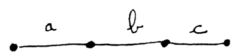
4) Indiquez les tâches critiques et le chemin critique.

Corrigé :

1°) a) Le niveau des tâches. C'est l'ordre dans lequel nous construisons les tâches.
N0 : Ensemble des sommets de niveau 0 : ce sont les sommets n'ayant pas de précédent. $N_0=\{c,d,e\}$.
N_1 *: On raye les sommets de niveau 0 partout où ils figurent dans la colonne des précédents P(x). Les sommets correspondant aux lignes entièrement rayées sont de niveau 1.*
$N_1=\{a,f,j\}$.
On réitère le processus $N_2=\{b,g,h,k\}$ $N_3=\{i\}$.

Tâches	Tâches antérieures P(x)
a	~~e~~
b	j,~~e~~
c	/
d	/
e	/
f	~~d~~
g	f,~~d~~
h	a,~~c,d~~
i	h,a,~~c,e~~,k,~~d~~,f
j	~~e~~
k	f,~~d~~

1°) b) Le tableau des tâches immédiatement antérieures.
Seules les tâches possédant plus d'un précédent (ou antécédent) sont à prendre en compte.
Ex :



Tâches	Tâches antérieures	Tâches immédiatement antérieures
c	a,b	b ,puisque a précède b

Dans l'exemple donné

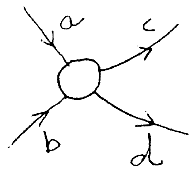
Tâches	Tâches immédiatement antérieures	
b	j	e précède j
g	f	d précède f
h	a,c,d	aucune relation d'antériorité entre a,c,et d
i	h,k	a,c,d,f précèdent h et e précède g
k	f	d précède f

2°) Construction du graphe

- *Tâches fictives*

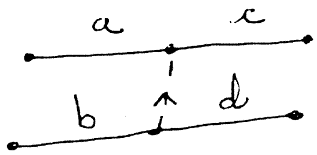
A chaque tâche correspond un arc du graphe dont la longueur est la durée de la tâche.
Chaque sommet du graphe est un événement ou étape.

Ex



«a et b sont terminées, c et d peuvent commencer»

Certaines relations d'antériorité nécessitent l'introduction de tâches fictives : si a et b précèdent c, b précède d, le schéma précédent n'est pas correct. On introduit donc une tâche fictive de la façon suivante :



Dans la pratique, pour réunir 2 tâches à une troisième, on crée 2 tâches fictives (Etape 1)

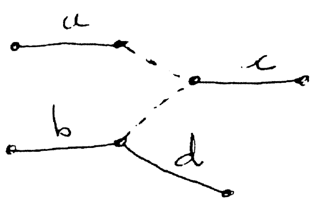


fig 1

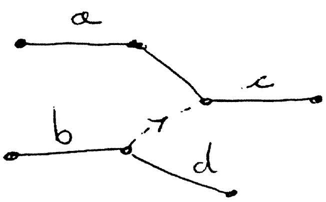


fig 2

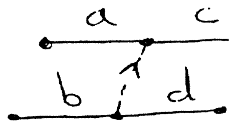


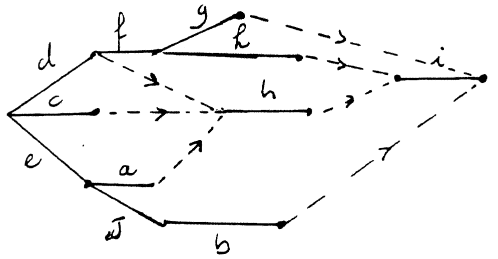
fig 3

Puis on supprime les tâches fictives inutiles (Etape 2)

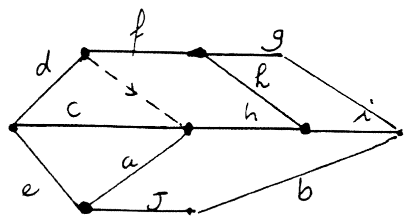
Une tâche fictive est inutile lorsque du nœud d'activité d'où elle est issue ne part aucune autre tâche. On supprime une tâche inutile en « tirant l'origine de cette tâche fictive jusqu'à ce qu'elle vienne coïncider avec l 'extrémité de cette même tâche.(fig2). On rectifie éventuellement le trajet a (fig 3)

Reprenons l'exercice donné et construisons le graphe initial.

Etape 1



Etape 2



- *Chaque étape est figurée par un cadre de ce type :*

t_x	t^*_x
x	

contenant
- *son numéro x*
- *sa date de début au plus tôt t_x*
- *sa date limite de réalisation t^*_x telle que la date au plus tôt de la fin des travaux ne soit pas retardée.*

Détermination des dates de début au plus tôt :

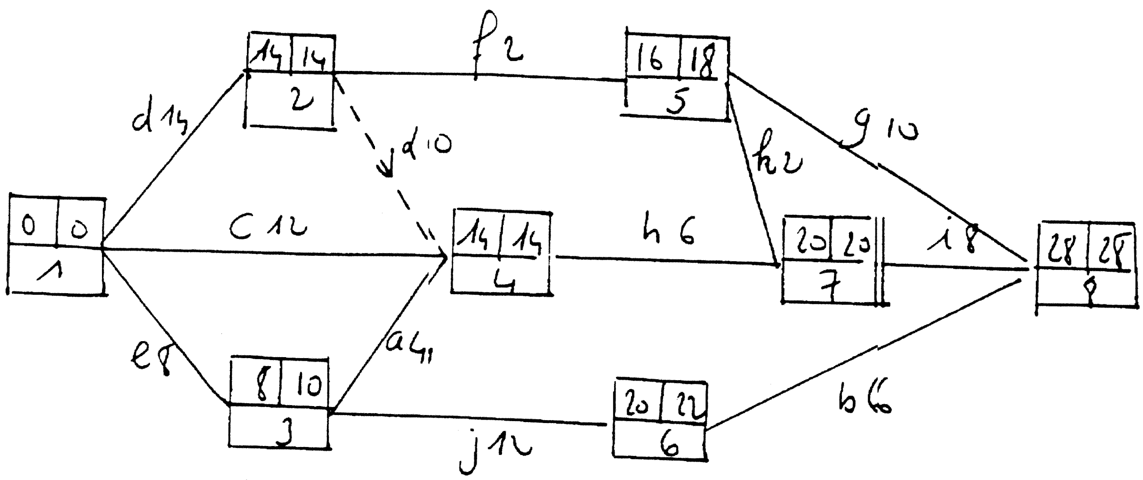
$t_x = Max_y\,(t_y + d_h)$ *d_h :durée de la tâche reliant x à y (y prédécesseur de x)*

t_x est donc la durée du chemin le plus long.

Détermination des dates limites de réalisation

$t^*_x = Min_y\,(t^*_y - d_h)$ *d_h :durée de la tâche reliant x à y (y successeur de x)*

On obtient ainsi le graphe définitif suivant :



Le temps minimum nécessaire au lancement du produit est de 28 jours.

3) Caractéristiques :tableau des dates de début au plus tôt et des dates de début au plus tard des diverses tâches

Tâches h	Durées d_h	Début au plus tôt $t_h=t_x$	Début au plus tard $t^*_h=t^*_x-d_h$	Marge totale $t^*_h-t_h$
a	4	8	10	2
b	6	20	22	2
c	12	0	2	2
d	14	0	0	0
e	8	0	2	2
f	2	14	16	2
g	10	16	18	2
h	6	14	14	0
i	8	20	20	0
j	12	8	10	2
k	2	16	18	2

h étant une tâche issue du sommet x, sa date de début au plus tôt est $t_h=t_x$.
h étant une tâche aboutissant au sommet x, sa date de début au plus tard est $t^*_h=t^*_x-d_h$.

La marge totale d'une tâche (ou marge de flottement) est la différence entre la date de début au plus tard et la date de début au plus tôt.

4°) Tâches critiques et chemin critique.

Une tâche est critique si tout retard dans sa réalisation entraîne un retard dans la réalisation du projet, soit si sa date de début au plus tôt est égale à la date de début au plus tard soit si sa marge totale est nulle.
Ainsi d,h,i sont des tâches critiques.

Le chemin critique est constitué des tâches critiques
d,[α],h,i est le chemin critique :
c'est le chemin le plus long (14+0+6+8=28)

(Comparer avec les longueurs des chemins c,h,i (12+6+8=26)
e,j,b (8+12+6=26)
e,a,h,i (26)
d,f,g (26)

Il indique le temps minimum nécessaire pour la réalisation du projet.

Vu à l'examen.

**** Exercice VII.2. Mathématiques Appliquées (BTS Comptabilité 1986)**

Une importante société de magasins alimentaires à grande surface diversifie son activité en créant des commerces dans de petites villes. La société crée le fonds de commerce qui est, ensuite, géré de façon autonome par un commerçant franchisé. La société réalise tout d'abord une étude d'implantation : étude de marché sur un certain rayon d'action et choix de la localité où sera installé le commerce.

<u>Première partie.</u>

A partir du jour où l'étude d'implantation est terminée, les tâches suivantes doivent être exécutées/

Tâches	Nature	Durée (en jours)	Tâches qui doivent être exécutées avant
A	Recherche d'un local	50	
B	Recherche d'un franchisé	45	
C	Constitution du dossier bancaire du franchisé	15	A,B
D	Constitution du dossier à la chambre de Commerce pour les inscriptions obligatoires	10	A,B
E	Formation du franchisé	30	B
F	Aménagement, plâtrerie, peinture du magasin	20	A
G	Réfection façade, enseigne	8	A
H	Equipement chambre froide	8	A,F
I	Equipement rayonnages	5	A,F
J	Implantation du magasin (disposition des articles)	6	A,B,E,F,G,H,I
K	Tirage en imprimerie des feuillets publicitaires	6	A,B,D
L	Distribution des feuillets publicitaires	2	A,B,D,K
M	Liste et envoi des invitation pour l'inauguration	6	A,B,D
N	Inauguration du magasin	1	toutes les autres

1°) Représenter cette succession de tâches par un graphe.

2°) Dire au bout de combien de jours au minimum le magasin peut être ouvert à la clientèle.

3°) Indiquer quel est le chemin critique.

4°) Préciser à quelles dates au plus tard devront commencer les tâches qui ne font pas partie du chemin critique.

Deuxième partie.

La durée de la plupart des tâches peut être considérée comme fixe. Néanmoins, si on considère l'expérience de la société dans la recherche du franchisé, on remarque que :

La durée moyenne de recherche du franchisé est de 18 jours.

Il y a 47% de chances que la recherche dure entre 28 et 48 jours.

1°) Si la durée de recherche du franchisé est une variable aléatoire qui suit une loi normale, quels sont les paramètres de cette loi ?

2°) En tenant compte du fait que le franchisé, une fois sélectionné, suit une formation de 30 jours, quelle est la probabilité que le franchisé ne soit pas prêt pour l'implantation du magasin qui est prévue 78 jours après le début des recherches ?

Corrigé :

1ère partie :

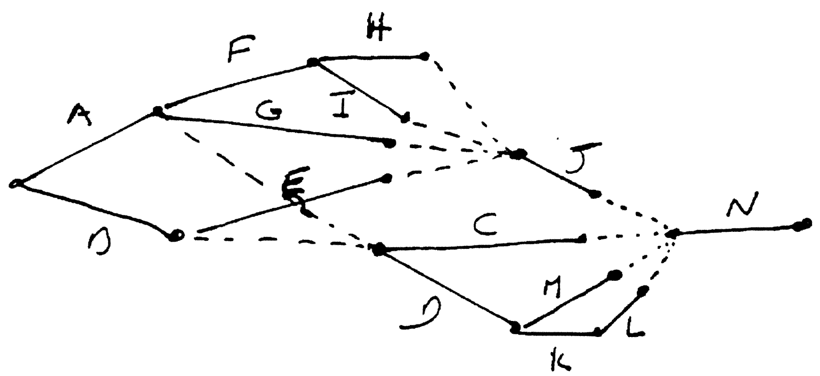
1) Niveau des tâches.

$N_0=\{A,B\}$ $N_1=\{C,D,E,F,G\}$ $N_2=\{H,I,K,M\}$ $N_3=\{J,L\}$ $N_4=\{N\}$

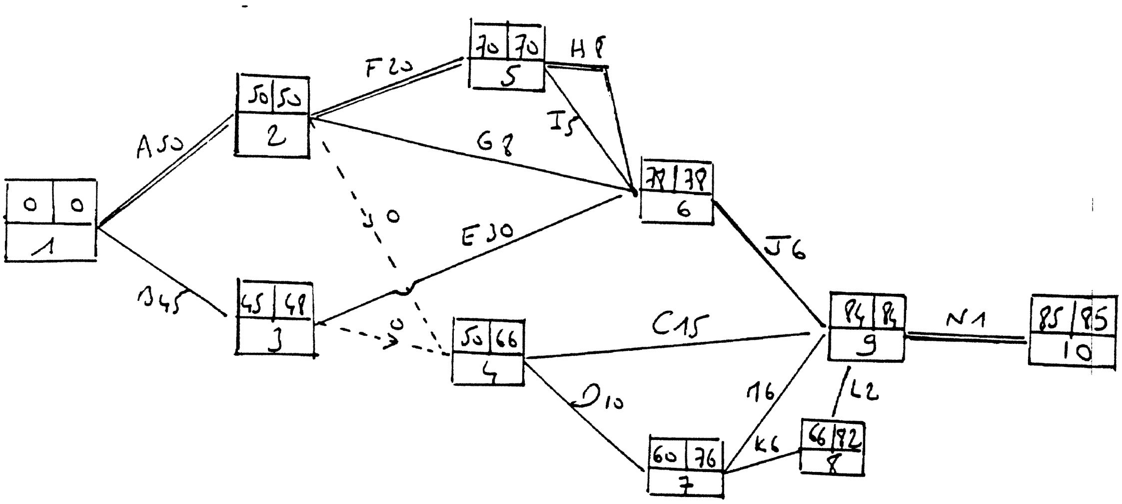
Tableau des tâches immédiatement antérieures.

Tâches	*Tâches immédiatement antérieures*
A	
B	
C	*A,B*
D	*A,B*
E	*B*
F	*A*
G	*A*
H	*F*
I	*F*
J	*E,G,H,I*
K	*D*
L	*K*
M	*D*
N	*C,J,L,M*

1er Schéma :



2ème Schéma : après suppression des tâches fictives inutiles, on obtient le graphe PERT suivant :



2°) Le magasin sera ouvert à la clientèle dans 85 jours au minimum.
3°) Le chemin critique est A,F,H,J,N.
(Pour les tâches critiques A,F,H,J,N, les dates de début au plus tôt coïncident avec les dates de début au plus tard.)
4°) Calendrier des dates au plus tôt et au plus tard de chaque tâche.

Tâches	Durées	Dates au plus tôt	Dates au plus tard	Marges totales
A	50	0	0	0
B	45	0	3	3
C	15	50	69	19
D	10	50	66	16
E	30	45	48	3
F	20	50	50	0
I	8	50	70	20
H	8	70	70	0
I	5	70	73	3
J	6	78	78	0
K	6	60	76	16
L	2	66	82	16
M	6	60	78	18
N	1	84	84	0

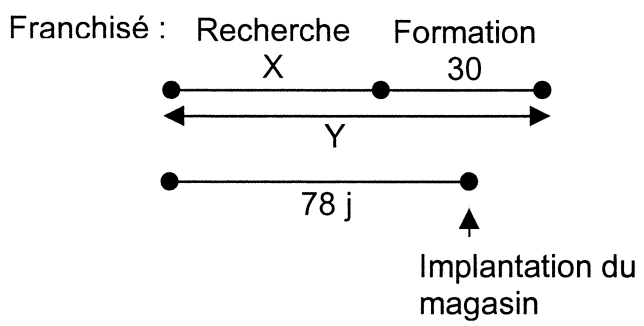
2[ème] partie :
X :VA durée de recherche du franchisé.
$X \sim N(m,\sigma)$ avec m=38, $P(28<X<48)=0 ;47$.

Détermination de l'écart type σ.
$P(28<X<48) = P(t1<T<t2) = G(10/\sigma)-G(-10/\sigma) = 2G(10/\sigma)-1 = 0,47$.

$t1 = (28-38)/\sigma = -10/\sigma$
$t2 = 10/\sigma$.

On déduit $G(10/\sigma) = 0,735$ d'où $10/\sigma = 0,62801$ et $\sigma = 15,92$ au centième près.

2°) Soit Y la VA durée de recherche du franchisé augmenté de la formation du franchisé :
$Y=X+30$.
Probabilité que le franchisé ne soit pas prêt pour l'implantation du magasin :
$P(Y > 78) = P(X > 48) = P(T > t1) = 1-P(T \leq t1) = 1-G(t1)=O,265$ puisque
$t1 = (48-38)/\sigma = 10/\sigma$ et $G(t1) = G(10/\sigma) = 0,735$



VIII Arbres, arborescence.

- **Exercices.**

Exercices.

Exercice VIII.1

: Que penser du problème du plus court chemin dans une arborescence?

Solution : Dans une arborescence il existe un ***unique*** *chemin de la racine vers un sommet donné. Il est donc inutile de chercher à calculer le plus court chemin de la racine à un autre sommet.*

Exercice VIII.2:

Soit G une arborescence donnée sous la forme d'un tableau père.
Donner des algorithmes permettant de:
1°) Reconnaître la racine.
2°) Trouver la racine.
3°) Reconnaître une feuille
4°) Afficher les ancêtres d'un sommet.
5°) Trouver le dernier ancêtre commun à 2 sommets donnés.

Solution:
1°) Père(i)=i.
2°)

> ***Algorithme*** *Trouver la racine.*
> ***Donnée:*** *Une arborescence fournie par le tableau père.*
> ***Résultat*** *La racine.*
>
> *Choisir un sommet i quelconque.*
> *Tant que père(i) n'est pas égal à i*
> *i=père(i).*
> *i est la racine.*

Complexité: profondeur de l'arborescence.

3°) Parcourir le tableau père de 1 à n à chaque fois marquer le père. Les sommets non marqués à la fin du parcours sont les feuilles.
Complexité O(n).

4°)

> ***Algorithme*** *Affiche les ancêtres.*
> ***Donnée:*** *Une arborescence fournie par le tableau père.*
> *et un sommet x*
> ***Résultat*** *La liste des ancêtres de x.*
>
> *Répéter jusqu'à x=père(x)*
> *x=père(x)*
> *afficher x*

Complexité: hauteur de l'arborescence.

5°)

Algorithme *Dernier ancêtre commun.(1ère version)*
Donnée; *2 sommets x et y d'une arborescence.*
Résultat *le dernier ancêtre commun à x et à y.*

Marquer x
Tant que n'est pas la racine
x=père(x)
Marquer x.
Tant que y n'est pas marqué
y=père(y)
Retourner y comme résultat.

Algorithme *Dernier ancêtre commun.(2ème version)*
Donnée; *2 sommets x et y d'une arborescence.*
Résultat *le dernier ancêtre commun à x et à y.*

Tant que x et y ne sont pas marqués
Marquer x; Marquer y
x=père(x); y père(y).
Retourner le sommet marqué.
Si x et y sont marqués tous les deux
Retourner celui qui n'est pas la racine.

Annexe.

Les programmes (extraits)

GRAPHES

Cette initiation aux graphes orientés doit être menée en étroite concertation avec les enseignements de l'informatique et de la gestion où cette étude est poursuivie.
L'objectif est d'introduire et de mettre en oeuvre, dans des situations concrètes très élémentaires et sans théorie générale, des algorithmes permettant de résoudre des problèmes figurant dans la rubrique des travaux pratiques.

Modes de représentation d'un graphe orienté : représentation géométrique, tableau des successeurs, matrice adjacente (booléenne). Chemin, circuit, boucle, chemin hamiltonien.	La définition d'un graphe orienté n'est pas au programme.
Arborescence.	La notion de connexité étant hors programme, on se limitera à la présentation d'exemples simples d'arborescence à partir de leur représentation géométrique, sans recherche d'une caractérisation générale.
Longueur d'un chemin, chemin optimal.	On observera l'importance du résultat: tout sous-chemin d'un chemin optimal est optimal.

Travaux pratiques

Exemples de mise en oeuvre d'algorithmes permettant d'obtenir pour un graphe : - les chemins de longueur p, - la fermeture transitive, - les niveaux, dans le cas d'un graphe sans circuit - les chemins de valeur minimale (ou le cas échéant de valeur maximale)	A partir d'exemples très élémentaires et sans introduire une théorie générale, on montrera l'intérêt des méthodes matricielles mettant en oeuvre l'addition et la multiplication booléennes des matrices adjacentes. Dans une évaluation en mathématiques, tout énoncé relatif à ces algorithmes doit comporter des indications sur la méthode à suivre.
Exemples de résolution de problèmes d'ordonnancement par la méthode des potentiels ou la méthode PERT.	Il s'agit d'un premier contact avec des méthodes largement utilisées en gestion ; ces méthodes ne peuvent faire l'objet <u>d'aucune évaluation en mathématiques</u>.

Bibliographie.

A.V.Aho,J.E.Hopcroft,J.D.Ullman, *The Design and Analysis of Computer Algorithms,* Addison-Wesley (1974)

C.Berge, *Graphes et hypergraphes*, Ed Dunod,(1973)

J.A.Bondy and U.S.R.Murty *Graph Theory with Applications*, North-Holland, New York, (1979)

S.A Cook, *The Complexity of theorem-proving procedures*, Proc. 3rd Annual ACM Symposium on Theory of Computing,(1971).

F.Droesbeke, M. Hallin, C.Lefevre Les graphes par l'exemple

M.R.Garey et D.S.Johnson, *Computers and Intractability: a guide to the Theory of NP-Completeness*, Freeman,San Francisco, CA, (1979)

M. Gondran et M Minoux, *Graphes et algorithmes*, Ed Eyrolles 1994.

C Goujet et C. Nicolas ; *Mathématiques Appliquées à la Gestion* ; Ed. Masson.

Roseaux (nom collectif), *Exercices et problèmes résolus de recherche opérationnelle, Tome 1 graphes: leurs usages, leurs algorithmes*, Ed MASSON (1986)

Aimé SACHE (nom collectif) *La théorie des graphes* Que sais-je (1974) P.U.F

R.Sedgewick, *Algorithms* second edition, , Addison-Wesley

LES PUBLICATIONS DU THEME 3 pour 1997-1998

de l'ADIREM et de la DIRECTION

DE L'ENSEIGNEMENT SCOLAIRE

(A 11)

1°) Pour les baccalauréats professionnels

. Baccalauréats professionnels 1997 du secteur tertiaire;
. Baccalauréats professionnels 1997 du secteur du bâtiment et de l'artisanat;
. Baccalauréats professionnels 1997 des métiers de l'électricité de la chimie, de l'énergétique;
. Baccalauréats professionnels 1997 des métiers de la maintenance et de la productique.

Ces publications s'adressent aux enseignants, aux auteurs de proposition de sujets d'examens, aux I.E.N....

Pour se les procurer, écrire à

L'IREM de MARSEILLE
Faculté des Sciences de Luminy
70, route Léon Lachamp - Case 901
13288 MARSEILLE Cedex

2°) Pour les BTS

. Les résultats d'une enquête sur les épreuves de mathématiques des BTS des filières électronique, mécanique, matériaux, laboratoire, tertiaire, arts graphiques de la session 1997 (2 fascicules).

. Quelques réflexions sur les énoncés de statistique et de probabilités, aux baccalauréats technologiques et aux Bts (octobre 1998).

. Quelques éléments de théorie des graphes pour le BTS Informatique de Gestion (mai 1998).

. Les plans d'expérience pour le BTS chimiste (mai 1998).

Pour se procurer ces publications, écrire à :

L'IREM de PARIS NORD
CII - LT
avenue J.B. Clément
93430 VILLETANEUSE