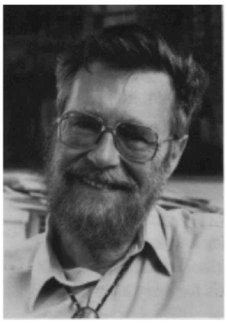
Chapitre 8: Graphes et optimisation

8.1 Un exemple en guise d'introduction

Introduction : *C'est l'histoire du livreur de pizza, contraint de livrer ses commandes en moins d'une demi-heure avec pour seule arme un scooter faiblard et poussif. C'est aussi l'histoire de monsieur Dupont qui, sa valise à la main, se rend à la gare de Lausanne, direction St-Moritz, sans bien savoir par où passer. C'est encore l'histoire d'Internet qui permet le transfert de tous types de données dans le monde entier via... quelle route au fait ?*

Dans les trois cas, la question que se posent les protagonistes avant de foncer tête baissée est la même : quel est le plus court chemin pour aller de mon point de départ à mon point d'arrivée sachant que je ne peux emprunter que des lignes ferroviaires, des routes, des réseaux ?

Qui dit trajet dit problème des plus courts chemins. Qui dit problème dit nécessité de trouver une solution !



Edsger W. Dijkstra (1930 – 2002)

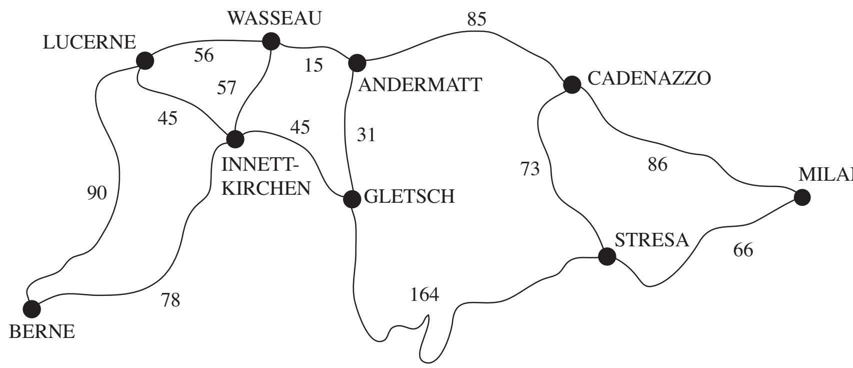
Le problème est donc posé : étant donné un ensemble de points (les gares, les serveurs) et un ensemble de liaisons entre ces points (les lignes ferroviaires, les routes) caractérisées par un certain poids (longueur en kilomètres ou, pourquoi pas, coût, nombre de tournants, débit, ...), comment trouver le chemin le plus court, le moins cher, le plus agréable entre deux points ?

*Il est difficile de parler d'algorithme naïf lorsqu'on s'attaque au problème des plus courts chemins, car quoi qu'il arrive, il faudra trouver un moyen de comparer différents chemins entre eux, de ne pas en oublier, bref de définir une façon de parcourir le graphe, ce qui, même à la main, est loin d'être instinctif. Nous en étudierons deux : l'**algorithme de Dijkstra** datant de 1959 et l'**algorithme MPM.***

Problème : Une entreprise de transport *ROUTEXPRESS* dont le siège est situé à BERNE, doit effectuer de fréquentes livraisons à MILAN, en dehors de la période hivernale.
Ses véhicules doivent donc traverser le massif des Alpes ; leur gabarit interdit l'usage des tunnels ferroviaires (Simplon, Saint-Gothard, Lötschberg, …) qui, sinon, faciliteraient le voyage.

Vu la fréquence et le coût de ces livraisons, l'entreprise désire déterminer l'itinéraire le plus court de BERNE à MILAN.

Démarche : D'une carte routière, on a extrait le plan suivant, indiquant les différents tracés possibles ainsi que les distances entre les villes (en km) :



Définitions

- On appelle **graphe pondéré**, un graphe dont les arêtes ont été affectées d'un nombre appelé **poids.**

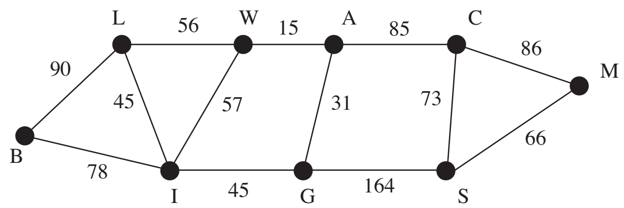
Dans les exemples étudiés ici, les poids affectés à chaque arête seront toujours positifs. Cette condition est assez banale lorsque les poids représentent par exemple des coûts, des distances, ou des temps. Elle n'est pas toujours réalisée lorsque par exemple les poids représentent des flux.

- **Poids d'un chemin :** C'est la somme des poids des arêtes qui constituent le chemin. On parlera aussi, selon le contexte, de **longueur du chemin.**

- **Plus court chemin d'un sommet à un autre :** C'est, de tous les chemins qui relient deux sommets, celui de longueur minimale.

Remarques: • *A priori, ce chemin n'est pas unique.*
• *On définirait de même le chemin* ***le plus long*** *(pour des graphes sans cycle).*

Démarche générale : *Notre problème s'énonce maintenant ainsi : dans le graphe de la figure suivante, déterminer le plus court chemin reliant le sommet B à M.*



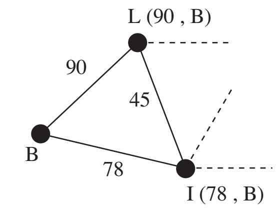
Le nombre de chemins reliant ces deux villes est limité, du moins si l'on élimine les chemins contenant des cycles, qui ne peuvent être minimaux. On peut en dresser la liste, calculer leur poids respectif, et déterminer la solution du problème :

Intuitivement, on ne procédera pas ainsi, mais "par étapes successives". Par exemple, on constatera que le passage par Innettkirchen pour Wasseau, qu'il soit ou non retenu finalement, est plus court que celui par Lucerne. Le chemin Berne–Lucerne–Wasseau peut ainsi être éliminé rapidement.

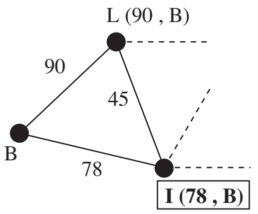
La règle très simple appliquée ici est à la base de l'algorithme de recherche du plus court chemin et peut être écrite ainsi :
Un plus court chemin [x_0 ; x_n] entre deux sommets x_0 et x_n d'un graphe est constitué des plus courts chemins reliant deux sommets du chemin [x_0 ; x_n]. (En d'autres termes : les sous-chemins des plus courts chemins sont des plus courts chemins. Cette règle, appelée parfois « principe d'optimalité », se démontre sans difficulté par l'absurde).

La recherche du plus court chemin entre deux sommets x_0 et x_n d'un graphe passe alors par les recherches successives des plus courts chemins reliant x_0 à tous les sommets du graphe susceptibles de se trouver sur le trajet. Reste à choisir ***dans quel ordre*** *on liste les sommets intermédiaires.*

1ère étape:



On part de Berne. On peut atteindre les deux villes L et I. On leur associe, entre parenthèses, la distance depuis Berne et l'initiale du sommet prédécesseur, ici B.

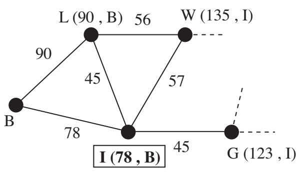


La distance la plus courte inscrite à cette étape est 78 km (si l'on excepte, bien entendu, 0 km pour Berne). Aucune autre ville n'étant accessible depuis Berne pour une distance plus courte, on peut assurer **qu'aucun autre trajet, passant par une autre ville, ne relie Berne à Innettkirchen pour une distance plus courte :** 78 km est la distance minimum d'un trajet Berne-Innettkirchen. Cette distance est notée **L(I)**. Innettkirchen sera dite **marquée (définitivement).**

2ème étape:

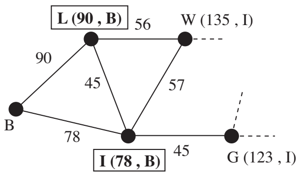
Notre voyageur va alors considérer les villes accessibles depuis Innettkirchen. Pour chacune, trois cas peuvent se présenter :

- *Cette ville X n'était pas provisoirement marquée : on la marquera provisoirement de la distance (L(I) + poids de l'arête δ(I ; X)).*
- *Cette ville X était déjà provisoirement marquée, mais la distance (L(I) + poids de l'arête δ(I ; X)) est inférieure à la distance du marquage provisoire : on la remplace.*
- *Cette ville X était déjà provisoirement marquée, et la distance (L(I) + poids de l'arête δ(I ; X)) est supérieure à la distance du marquage provisoire : on garde la distance provisoire.*



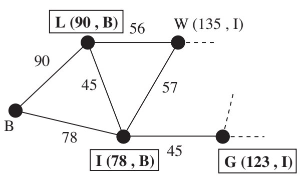
Dans notre exemple :

La distance pour L reste inchangée (90 < 78 + 45),
W est marqué provisoirement d'une distance 135 (78 + 57),
G est marqué provisoirement d'une distance 123 (78 + 45).



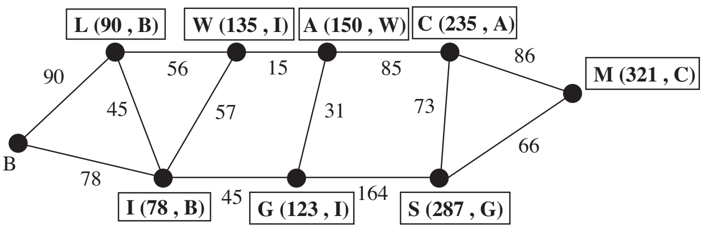
À ce stade, parmi les villes provisoirement marquées, L admet la plus petite distance. On marquera alors définitivement L(L) = 90.

3ème étape:



À présent, on marque la ville adjacente à L en constatant que l'on ne modifie pas le marquage provisoire car 135 < 90 + 56. G admettant la plus petite distance, on la marque définitivement.

On réitère ainsi ce processus jusqu'à atteindre la ville de Milan. Nous obtiendrons alors le graphe suivant :



On reconstitue le chemin voulu en partant de Milan et en remontant étape par étape le chemin à l'envers à l'aide des sommets prédécesseurs: M-C-A-W-I-B.

Solution : Il s'agit donc du trajet de 321 km :
Berne – Innetkirchen – Wasseau – Andermatt – Cadenazzo – Milan.

Exercice 103 Reprendre la démarche <u>complète</u> de l'exemple Berne – Milan.

8.2 L'algorithme de DIJKSTRA

Formulons la méthode utilisée par l'algorithme suivant :

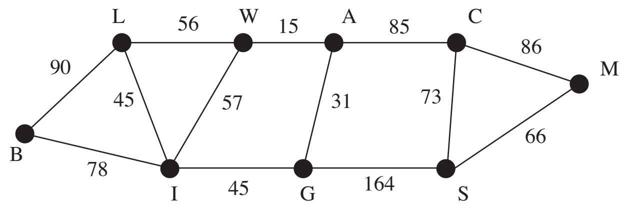
<u>**Algorithme du plus court chemin (Dijkstra)**</u>

Donnée : Un graphe simple, non orienté, pondéré, à n sommets
Résultat : Le plus court des chemins d'un sommet de départ à tous les autres sommets.

*** Initialisation ***
S = Ø ; $L(x_1) = 0$ et $L(x_i) = \infty$ pour $i \neq 1$
*** Itération ***
Répète
  choisis un sommet $x_i \in V\text{-}S$ avec $L(x_i)$ minimal ;
  si $L(x_i) = \infty$, **alors** halte ;
  ajoute le sommet x_i à l'ensemble S ;
  si $S = V$, **alors** halte ;
  pour chaque voisin $x_j \in V\text{-}S$ du sommet x_i,
    si $L(x_j) > L(x_i) + \partial(x_i , x_j)$
      remplace $L(x_j)$ par $L(x_i) + \partial(x_i , x_j)$
      et remplace $p(x_j)$ par x_i ;

On représentera les étapes de l'algorithme en complétant le graphe et à l'aide d'un tableau mentionnant toutes les étapes :

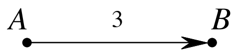
Résolution type
(à compléter ensemble)



B	L	I	W	G	A	S	C	M	
0	∞	∞	∞	∞	∞	∞	∞	∞	**B**

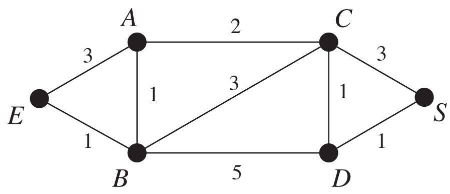
Dans le cas des graphes orientés ?

L'algorithme de DIJKSTRA peut facilement être adapté à un graphe orienté, en indiquant un poids de ∞ si l'arc n'est pas orienté dans le "bon sens".

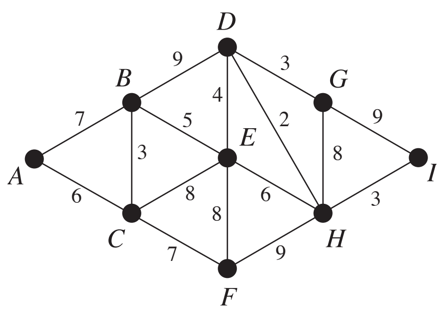


Poids de $[A\ ;\ B] = 3$
Poids de $[B\ ;\ A] = \infty$

Exercice 104 Appliquer l'algorithme de Dijkstra pour déterminer le plus court chemin entre E et S et proposer la résolution à l'aide du tableau.

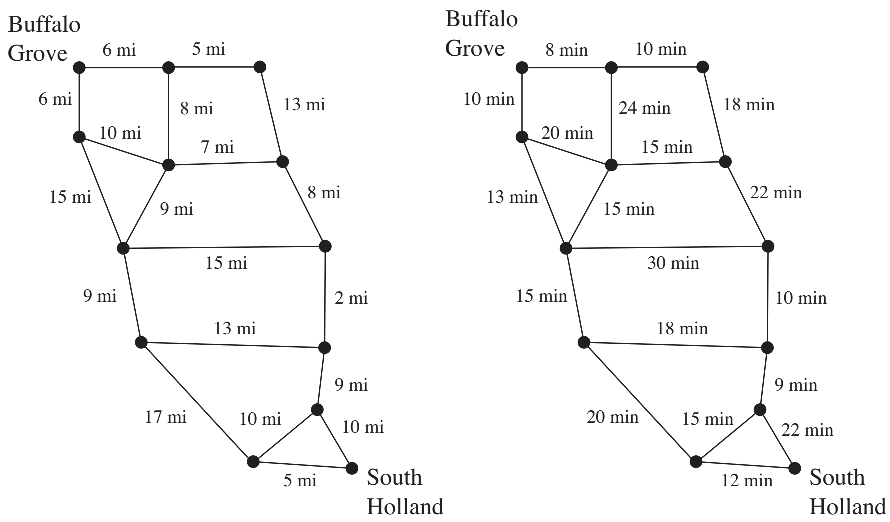


Exercice 105 Trouver le plus court chemin du sommet A vers tous les sommets en utilisant l'algorithme de Dijkstra.



Dans chaque cas, proposer un chemin minimum et indiquer le nombre de chemins minimums possible.

Exercice 106 Les deux cartes suivantes représentent ; à gauche : le trajet en miles du train *Expressways in the Chicago area* et à droite la durée des différentes étapes.



a) Utiliser l'algorithme de Dijkstra pour déterminer le plus court trajet de Buffalo Grove à South Holland.

b) Utiliser l'algorithme de Dijkstra pour déterminer le trajet le plus rapide pour relier les deux mêmes villes.

Exercice 107 La matrice qui suit donne en heures les durées des vols entre certaines villes $v_1, v_2, \ldots, v_6$:

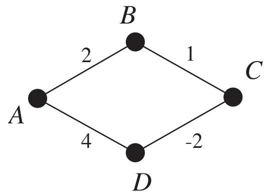
$$\begin{pmatrix} 0 & 3 & \infty & 5 & \infty & \infty \\ 3 & 0 & 5 & 2 & 4 & \infty \\ \infty & 4 & 0 & \infty & 4 & 3 \\ 6 & 2 & \infty & 0 & 4 & 4 \\ \infty & 4 & 4 & 5 & 0 & 2 \\ \infty & \infty & 5 & 4 & 3 & 0 \end{pmatrix}$$

Le terme (i,j) de cette matrice est égal à ∞ lorsque le vol au départ de la ville v_i à destination de v_j n'existe pas.

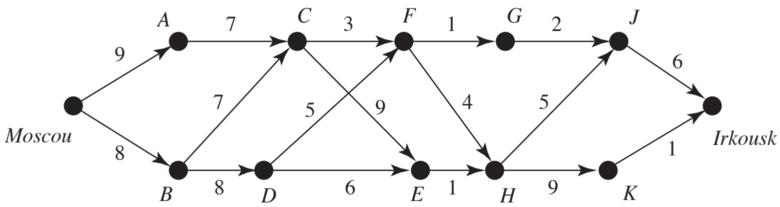
a) En ne tenant pas compte de la durée des escales, quel est l'itinéraire le plus rapide de v_1 à v_6.

b) S'il y a une escale obligatoire de respectivement 2, 3, 1, 1, 4, 5 heures aux villes $v_1, \ldots, v_6$, quel est alors l'itinéraire le plus rapide de v_1 à v_6 ?

Exercice 108 Dans le graphe ci-dessous contenant un poids négatif, montrer que l'algorithme de Dijkstra ne peut pas être appliqué.



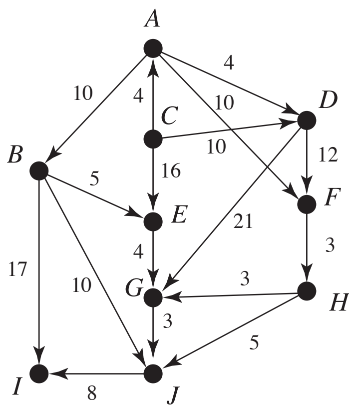
Exercice 109 Partant de Moscou, Michel Strogoff, courrier du tsar, devait rejoindre Irkousk. Avant de partir, il avait reporté sur une carte pour chaque liaison entre deux villes les « chances » de réussite. Ces chances étaient exprimées par un entier de 1 à 10 (mesurant le nombre de chances sur 10 de réussite). Ignorant le calcul de probabilités, il avait choisi son itinéraire en <u>maximisant</u> la somme globale des chances.



a) Déterminer le trajet de Michel Strogoff.

b) Quel aurait été son trajet s'il avait connu le calcul des probabilités ?

Exercice 110 Déterminer le plus court chemin de C à I



Exercice 111 L'un d'entre vous se sent-il prêt à démontrer que l'algorithme de Dijkstra fournit bien le plus court chemin d'un sommet initial à un sommet final ?

8.3 Problèmes d'ordonnancement

De quoi s'agit-il ?

Il s'agit de savoir planifier l'exécution de tâches qui ont une certaine durée, et qui ont entre elles des relations d'antériorité (par exemple, dans les révisions qu'il faut faire avant de passer le baccalauréat, il y a des chapitres qu'il faut revoir avant d'autres).

Exemple :

L'entreprise Oméga a procédé à la définition d'un certain nombre de tâches à effectuer. On a rempli la colonne « antériorité » avec les tâches qui doivent être exécutées avant celle considérée. Une évaluation du temps de chaque tâche a également été proposée.

Tâche	Durée en semaines	Antériorité
A	3	
B	4	A
C	5	A
D	2	A
E	2	B, C
F	3	D

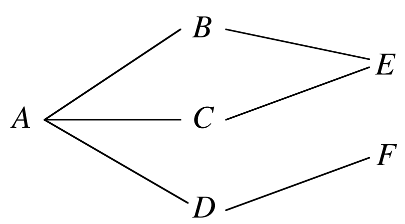
Proposer un agenda des différentes tâches, en indiquant le nombre de semaines minimum puis maximum à entrevoir avant la fin des différentes tâches.

Il existe plusieurs algorithmes possibles pour résoudre les problèmes d'ordonnancement. Les deux plus fréquemment utilisées sont la méthode PERT (méthode américaine) et la méthode MPM (méthode française datant de 1960). Concentrons-nous sur cette dernière.

Méthode française MPM (Méthode Potentiel Métra)

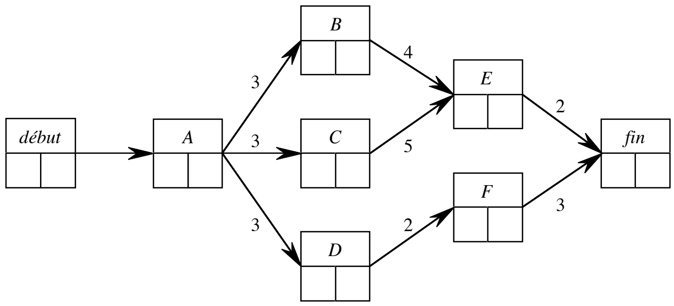
1) À partir du tableau, on réalise un premier graphe appelé graphe de niveau dont les sommets sont les tâches et qui permet de mettre en évidence les antériorités :

niveau 1 **niveau 2** **niveau 3**



2) On crée un graphe orienté dont les sommets sont les tâches ; on crée deux tâches fictives qui sont les tâches « début » et « fin » (sous-entendu « du processus »).

Les arcs sont les relations d'antériorité immédiate ; ils sont valués par la durée de la tâche source.

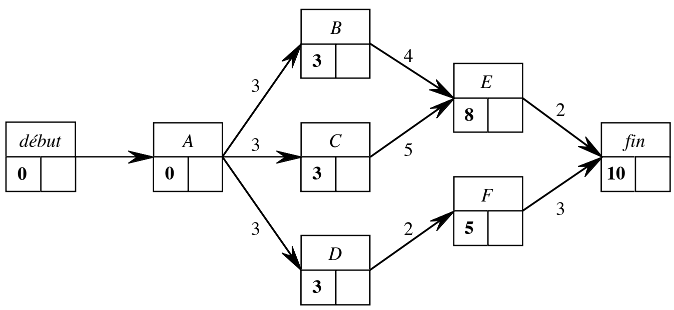


Chaque sommet comporte 3 zones qui contiennent respectivement :

<table>
<tr><td colspan="2">Nom de la tâche</td></tr>
<tr><td>Début au plus tôt</td><td>Début au plus tard</td></tr>
</table>

3) **<u>Dates « au plus tôt</u>** »

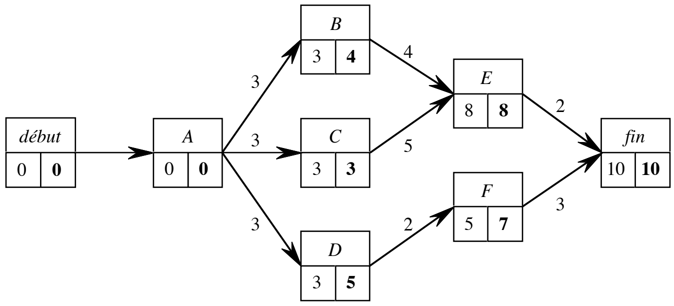
On traite les sommets par niveaux en partant du début.
Pour chaque sommet i on note la date t_i qui est la longueur du plus **long** chemin de la tâche initiale à la tâche i.



Le travail ne pourra donc pas être terminé avant 10 semaines.
La tâche E ne pourra pas commencer avant 8 semaines, la tâche F avant 5 semaines, etc.

4) <u>Dates « au plus tard »</u>

On traite les sommets en partant de la fin (en marquant **10** pour le sommet « fin »).
Pour chaque sommet on note la date t_i^* qui est la longueur du plus **court** chemin de la tâche *i* à la tâche « fin ».



Pour effectuer l'ensemble des tâches en 10 semaines, il faudra avoir commencé la tâche *E* au bout de 8 semaines, commencé la tâche *F* au bout de 7 semaines, etc.

5) <u>Exploitation de ce graphe</u>

Il y a des **tâches critiques**, celles pour lesquelles on a $t_i = t_i^*$: la tâche *E* devra obligatoirement débuter durant la 8$^{\text{ème}}$ semaine (ni plus ni moins) pour que le processus soit achevé au bout des 10 semaines. Les tâches critiques définissent un ou plusieurs **chemins critiques** composés de tâches dont l'exécution ne doit connaître aucun retard pour que le projet soit achevé au plus tôt.

Par contre, il y a de la latitude pour les tâches qui ne sont pas critiques : la tâche *F* pourra être démarrée entre la semaine 5 et la semaine 7.
De même, il y a un chemin critique : *A* – *C* – *E* – fin (il y a toujours un chemin critique dans un graphe MPM).

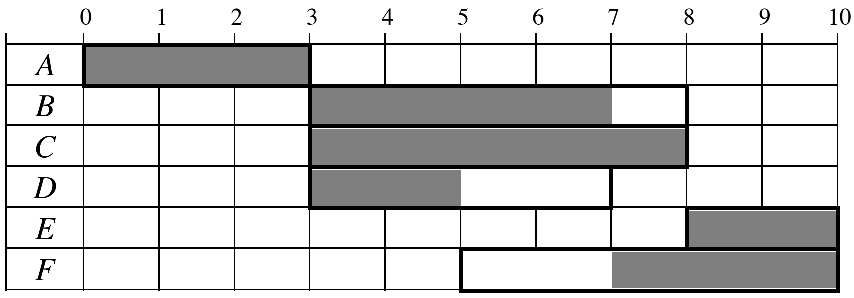
Définition : **Marge totale :** retard maximum que peut avoir une tâche sans retarder la fin du projet (les tâches critiques n'ont pas de marge). On l'obtient en calculant $t_i^* - t_i$.

Dans notre exemple, on obtient le tableau suivant :

Tâche	**Marge totale**
A	0
B	1
C	0
D	2
E	0
F	2

5) **Diagramme de Gantt**

Il s'agira de représenter graphiquement le déroulement du projet ; les tâches à effectuer sur les plages à disposition durant les 10 semaines (numérotées de 0 à 9).



Exercice 112 Proposer le diagramme de Gantt de la planification suivante :

Tâche	Durée en semaines	Antériorité
A	3	*B*
B	8	-
C	2	*A, B, F*
D	6	-
E	5	*B, D*
F	4	*B*

Exercice 113 La mise en exploitation d'un nouveau gisement minier demande la réalisation d'un certain nombre de tâches. Le tableau suivant représente ces différentes tâches avec leurs relations d'antériorité.

Tâche	Description	Durée (en jours)	Tâches antérieures
A	obtention d'un permis d'exploitation	120	-
B	établissement d'une piste de 6 km	180	*A*
C	transport et installation à pied d'œuvre de 2 sondeuses	3	*B*
D	création de bâtiments provisoires pour le bureau des plans, le logement des ouvriers sondeurs	30	*B*
E	goudronnage de la piste	60	*B*
F	adduction d'eau	90	*D*
G	campagne de sondage	240	*C,D*
H	forage et équipement de trois puits	180	*E,F,G*
I	transport et installation au fond du matériel d'exploitation	30	*J,H*
J	construction de bureaux et logements, ouvriers et ingénieurs	240	*E,F,G*
K	traçage et aménagement du fond	360	*J,H*
L	construction d'une laverie	240	*J,H*

a) Déterminer les dates au plus tôt et les dates au plus tard de chaque tâche.
b) Déterminer le temps minimum de réalisation de l'ensemble.
c) Proposer un chemin critique.
d) Préciser les marges totales de chaque tâche.

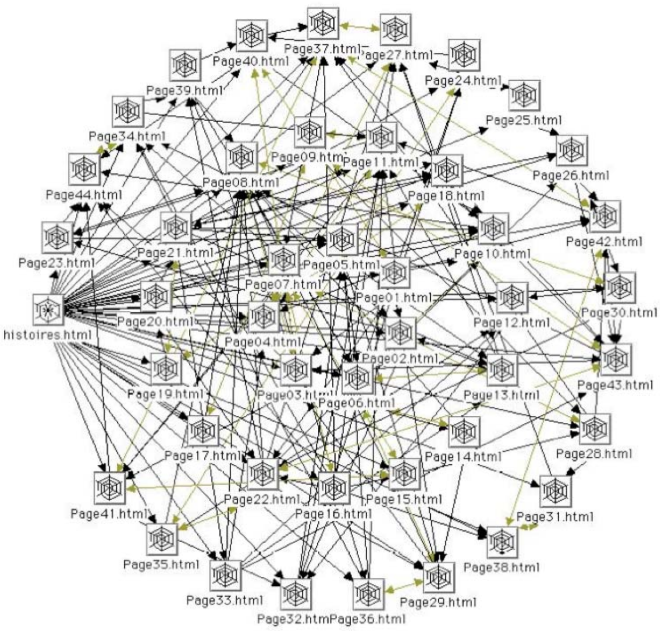
Exercice 114 Un producteur de cinéma est confronté au problème de planning de son prochain film selon les tâches suivantes :

Tâche	**description**	**durée (j)**	**antériorités**
A	écriture du scénario	30	-
B	choix et recrutement des comédiens	12	15 jours après le début de A
C	choix du lieu de tournage	8	20 jours après le début de A
D	découpage technique	4	A et C doivent être terminées
E	préparation des décors	7	C et D doivent être terminées
F	tournage des extérieurs	10	A, B, C et D doivent être terminées
G	tournage des intérieurs	12	D, E et F doivent être terminées
H	synchronisation	3	F et G doivent être terminées
I	montage	14	H doit être terminée
J	accompagnement sonore	7	ne peut commencer que 3 jours après le début de I et après la fin de H
K	mixage	6	I et J doivent être terminées
L	tirage de la copie zéro	1	ne peut commencer que 2 jours après la fin de K

a) Déterminer les dates au plus tôt et les dates au plus tard de chaque tâche.
b) Déterminer le temps minimum de réalisation de l'ensemble.
c) Proposer un chemin critique.
d) Préciser les marges totales de chaque tâche.

Conclusion et références

Nous voici au bout de notre voyage dans la théorie des graphes. Mais peut-être pas au bout du vôtre… Cette théorie récente cache encore quelques résultats à découvrir, quelques algorithmes à optimiser et quelques théorèmes à compléter. Que ce soit en informatique, en mathématiques, en sciences économiques, …, les outils de la théorie des graphes sont encore en pleine expansion.



Liens entre différentes pages html d'un site Internet

Quelques références et bibliographie :

- Kenneth H. Rosen. (1998). *Mathématiques discrètes*, Chenelière/McGraw-Hill
- Richard J. Trudeau. (1993). *Introduction to graph theory*, Dover
- La revue *Tangente* pour les annexes *5 chapitre 5* & *1 et 2 chapitre 7*
- Eric Sigward. (2002). *Introduction à la théorie des graphes*
  www.ac-nancy-metz.fr/enseign/maths/m2002/institut/ipr/graphes/Graphes.pdf
- http://icosaweb.ac-reunion.fr/RsrcPeda/General/Lycee/TheorieDesGraphes/Graphespage01.htm
- Didier Müller (2011). *Introduction à la théorie des graphes*
  www.apprendre-en-ligne.net/graphes/graphes.pdf