

Potentiels-tâches

Objectifs

- Ordonnancement par la méthode potentiels-tâches
- mise en oeuvre; l'interface QUI DEVRA IMPÉRATIVEMENT ÊTRE RESPECTÉE sera la suivante :

<nomprogramme> <fichier donnee> <fichier resultat>

Présentation du problème

Il faut réaliser un travail complexe; ce dernier peut être décomposé un certain nombre de tâches.

Il existe des contraintes d'ordre entre les tâches, certaines devant être terminées pour pouvoir en commencer de nouvelles. Si des tâches n'ont pas de contrainte d'antériorité (direct ou indirect), elles peuvent être exécutées en parallèle.

De plus, chaque tâche a une durée déterminée nécessaire à sa réalisation.

Le chef de projet souhaite établir un planning général avec une date de début pour chaque tâche garantissant une durée globale minimale de l'ensemble du travail.

Les différents éléments attendus sont :

- Pour chaque tâche :
 - son identifiant
 - sa date au plus tôt de démarrage
 - sa date au plus tard de démarrage
 - sa marge
- Le chemin critique : l'enchaînement des tâches qui ne doivent pas prendre de retard sous peine de retarder l'ensemble du travail; les tâches du chemin critique ont une marge nulle.

Principe de la méthode

Le travail est représenté sous la forme d'un graphe orienté construit de la manière suivante :

- à chaque tâche correspond un sommet (auquel seront associés ses attributs)
- de ce sommet partiront les arcs vers les autres sommets représentant les tâches nécessitant la terminaison de la tâche associée à celui-ci (vers ses successeurs, selon la relation d'antériorité).
- lors de la lecture des données, les arcs vers les prédécesseurs (données dans le fichier) seront aussi représentés dans la structure de donnée.

Pour simplifier, deux tâches fictives de durée nulles sont introduites afin d'avoir un début et une fin unique :

- *alpha* : tâche initiale de durée nulle, qui devient le prédécesseur commun de toutes les tâches qui n'en avaient pas
- *omega* : tâche finale de durée nulle qui aura comme prédécesseurs toutes les tâches qui n'avaient pas de successeur.

Ces deux tâches seront créées lors de la lecture des données et la construction du graphe.

Format des données

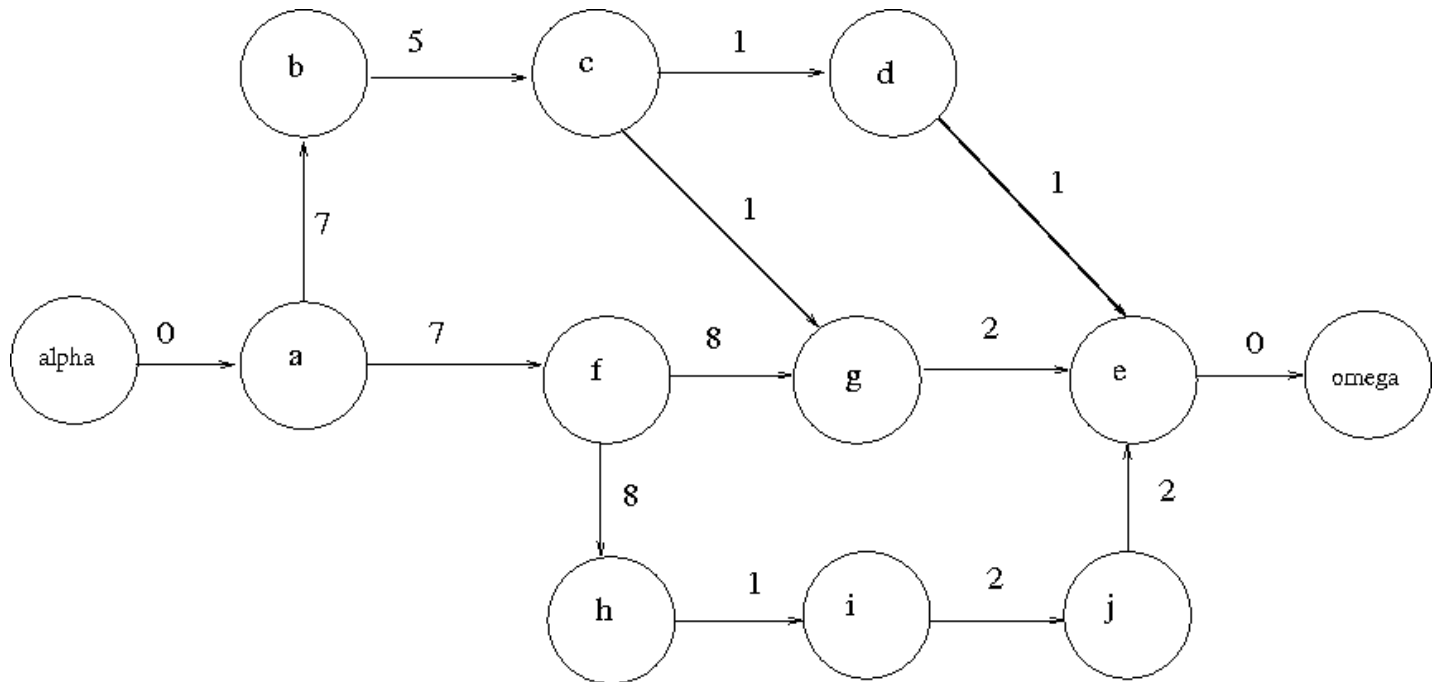
Les données du problèmes sont données sous la forme suivante :

- chaque ligne correspond à un sommet
- chaque ligne est constituée des éléments suivants :
 - identificateur du sommet (étiquette) - chaîne de caractères
 - durée - entier
 - degré : nombre de prédécesseurs (éventuellement 0) - entier
 - prédécesseurs : suite d'identificateurs (de longueur égale à degré).

Ainsi, pour le fichier suivant :

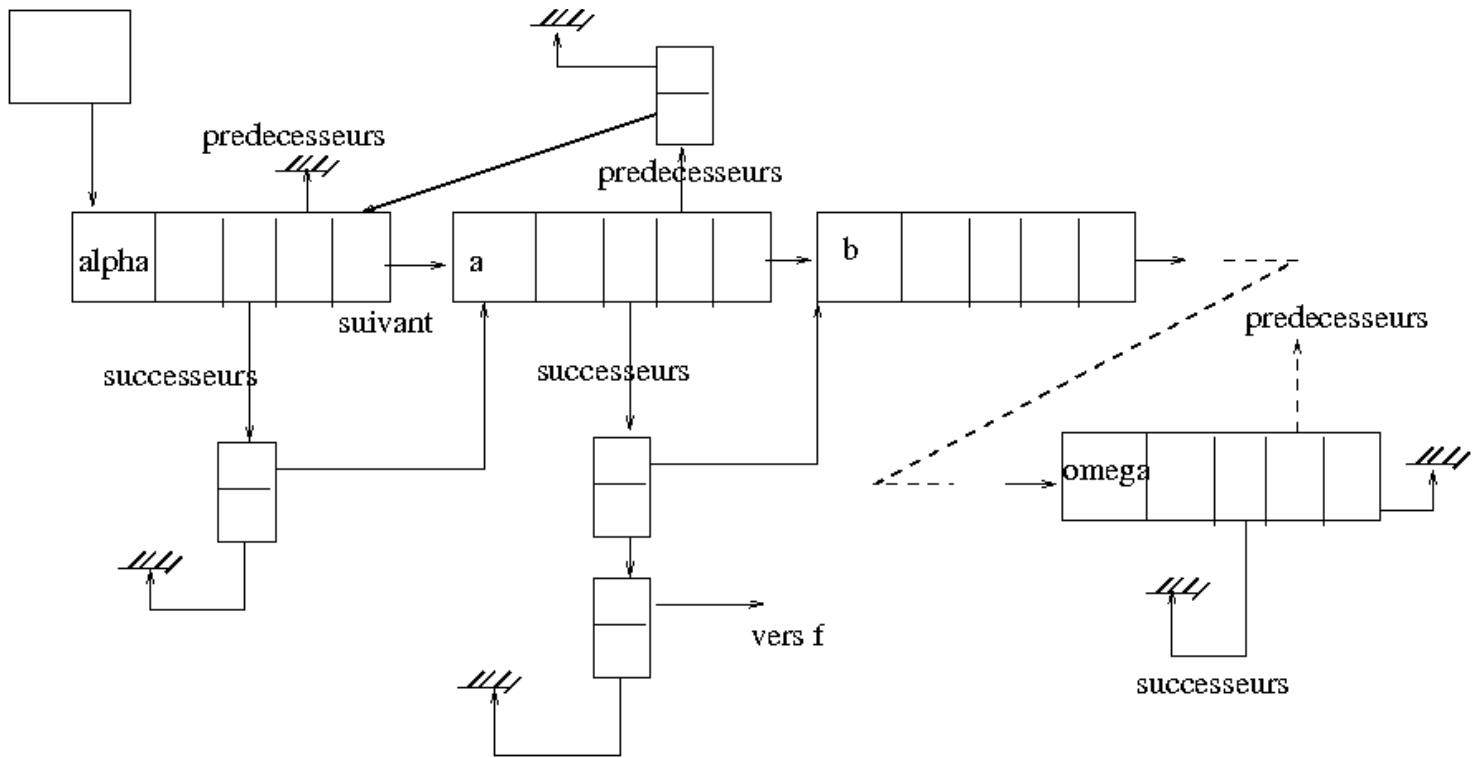
a	7	0	
b	5	1	a
c	1	1	b
d	1	1	c
e	1	3	d g j
f	8	1	a
g	2	2	f c
h	1	1	f
i	2	1	h
j	2	1	i

Le graphe suivant sera construit



En mémoire, une représentation simple pourrait être une structure de données chaînées :

- une liste constituant l'ensemble des sommets du graphes, la référence au premier élément de la liste représentant ainsi le graphe
- dans chaque cellule, qui représente un sommet, une liste des successeurs, le successeur étant représenté par un pointeur vers sa cellule. Il y a aussi la liste des prédécesseurs.



Décomposition du travail

- Lecture des données et construction du graphe
- ajout de la tâche fictive *omega*
- classement des sommets par rang (niveau) : mise en forme des données pour les calculs suivants
- calcul des dates de début au plus tôt (calcul de *alpha* vers *omega*)
- calcul des dates de début au plus tard (calcul de *omega* vers *alpha*)
- calcul des marges
- détermination du chemin critique

Détails des différentes étapes

Structure de donnée

La première étape consiste à définir dans le langage choisi les structures de données utilisées. Pour cela, il est conseillé de suivre les directives.

Lecture du fichier

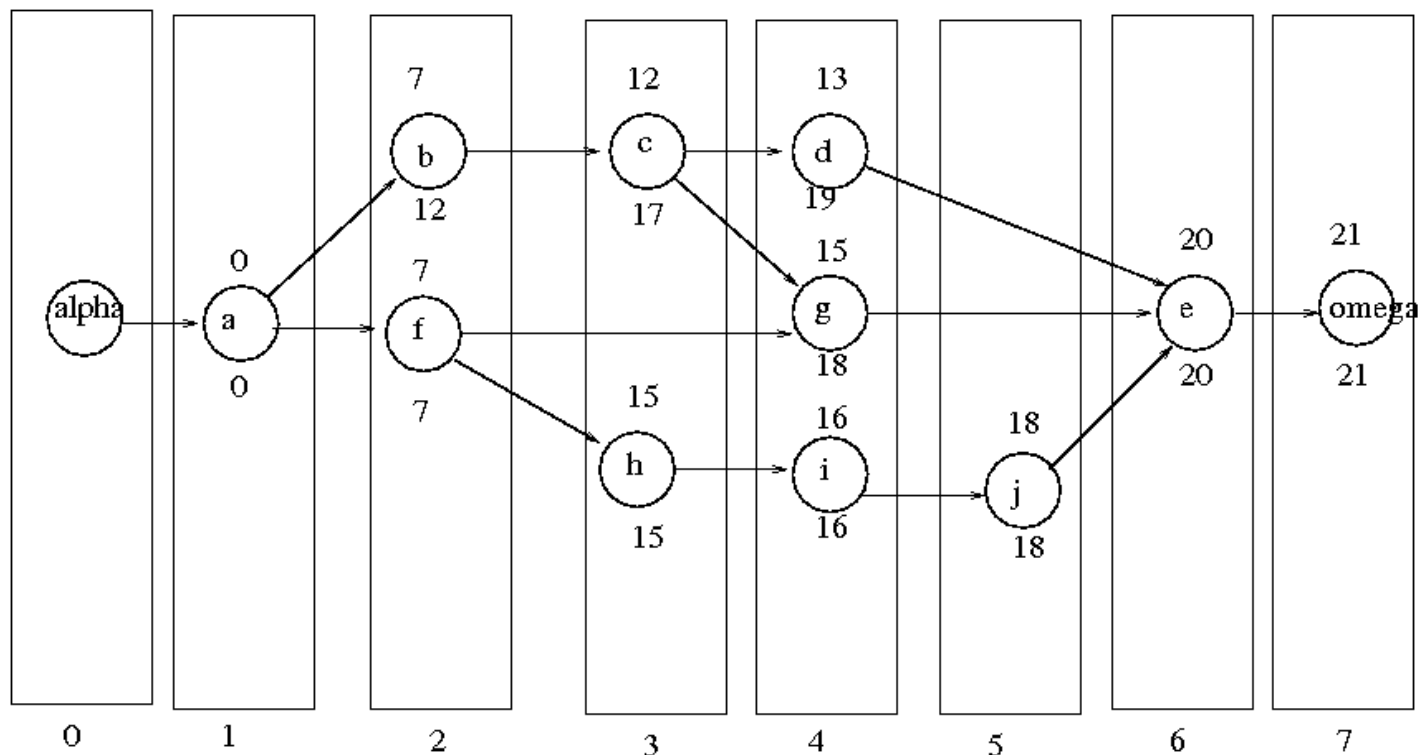
Le graphe sera construit lors de la lecture (une passe suffit). Une fonction que nous appellerons *place* peut s'avérer utile. Son comportement est le suivant : à partir d'un identificateur (lu dans le fichier), la fonction parcourt la liste à la recherche de la cellule correspondante; si celle-ci est déjà présente, son adresse est retournée; dans le cas contraire, la cellule est créée (la place n'a pas d'importance) et son adresse retournée.

Organisation par niveau (rang)

La structuration des données à l'issue de la phase de lecture n'est pas adaptée pour les traitements suivants. Une réorganisation des sommets par niveau est donc recommandée. La définition est la suivante :

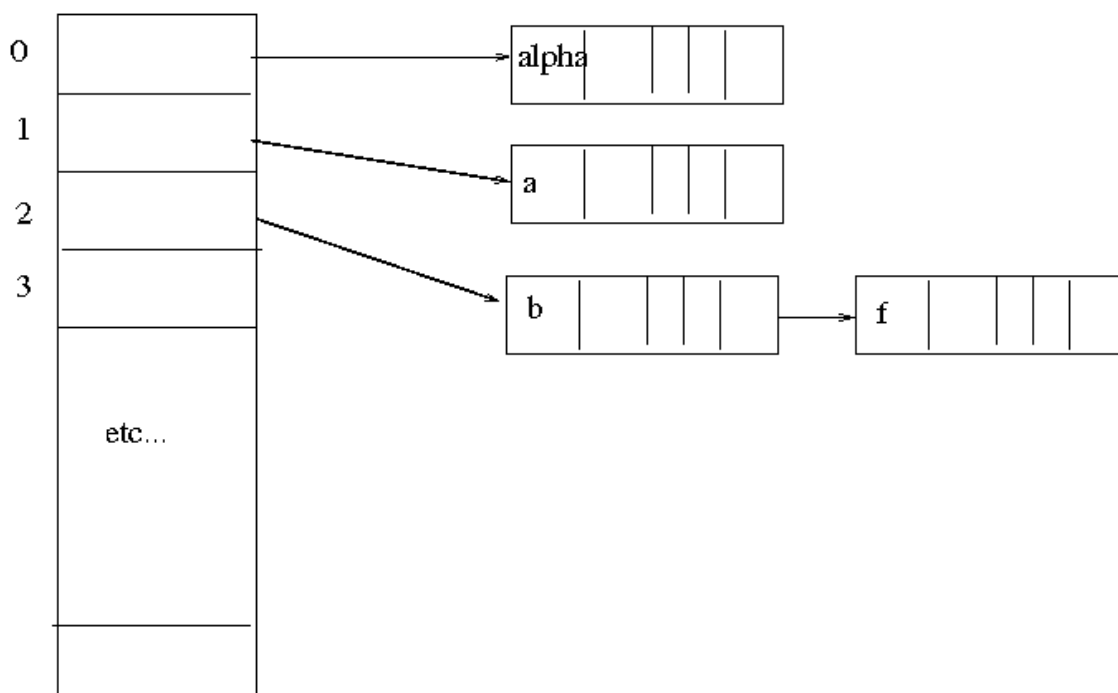
- le sommet de départ (unique) est de niveau 0
- un sommet de niveau i ne peut pas être à l'origine d'un arc vers un sommet de niveau j si $j \leq i$ (pas d'arc arrière)

Pour le graphe donnée en exemple, nous obtenons :



Nous proposons donc de modifier la structure en constituant des listes (ensembles) de sommets de même niveau; pour chaque niveau, nous aurons donc une tête de liste qui sera mémorisée dans un tableau.

Pour ce faire, le champs *degreeInterieur* pourra être utilisé; en effet, pour un sommet, si chaque fois que l'un de ses prédécesseurs est mis dans la table, son champ *degreeInterieur* est décrémenté, ce sommet pourra être mis dans le tableau (à l'entrée suivante) si le nombre de ses prédécesseurs est nul.



Calcul des dates au plus tôt

Le calcul sera fait dans le sens des niveaux croissants (la date au plus tôt de la première tâche - *alpha* - est 0). Pour chaque sommet, il faudra calculer pour tous les prédécesseurs la valeur (date au plus tôt + durée) et de choisir la valeur maximum comme date au plus tôt.

Calcul des dates au plus tard

À partir du niveau maximale (la marge de la tâche fictive *omega* est nulle), pour chaque tâche du niveau, sa date de début au plus tard est le minimum, sur tous ses prédécesseurs de la différence (date au plus tard du successeur - durée de la tâche).

Calcul des marge

Pour une tâche, la marge est définie comme la différence entre la date au plus tard et la date au plus tôt.

Tâche critique

Une tâche de marge nulle est dite critique.

Chemin critique

Toutes chaînes de tâches en *alpha* et *omega* constituent un chemin critique; il en existe au moins une.