

Esiee-Paris - Algorithmique 2 - rene.natowicz@esiee.fr
Ordonnancement optimal d'un projet - mercredi 27 mai 2015

Il s'agit de calculer le temps minimum de réalisation d'un projet.

Définitions :

Projet, tâches : un projet P est un ensemble de n tâches : $P = \{0, \dots, n-1\}$.

Prédécesseurs d'une tâche : chaque tâche i a un ensemble $pred(i)$ de tâches qui la précèdent directement : la tâche i ne peut commencer que si chacune des tâches de son ensemble $pred(i)$ a été réalisée. La tâche « début du projet » n'a pas de prédécesseur : $pred(0) = \emptyset$.

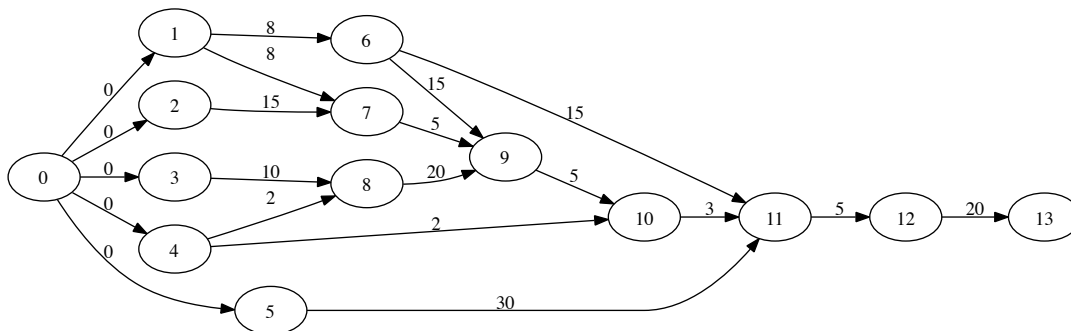
Durée d'une tâche : chaque tâche i a une durée notée d_i . Les tâches « début du projet » et « fin du projet » sont les tâches 0 et $n-1$. Leurs durées sont nulles : $d_0 = 0, d_{n-1} = 0$.

Projet réalisé : c'est un projet dont toutes les tâches ont été réalisées.

Parallélisme de réalisation des tâches : on dispose d'autant de ressources que nécessaire pour que les tâches soient réalisées avec le parallélisme maximum.

Exemple : le graphe ci-dessous représente le projet $P = \{0, \dots, 13\}$:

- ce projet a $n = 14$ tâches ;
- la tâche 10 est de durée $d_{10} = 3$. L'ensemble de ses prédécesseurs est $pred(10) = \{4, 9\}$. La tâche 10 ne peut donc commencer avant que chacune des deux tâches 4 et 9 ne soit réalisée, mais la tâche 10 peut commencer indépendamment de la réalisation de la tâche 5 ;
- la tâche 5 peut être réalisée en parallèle avec les tâches 1, ..., 4, 6, ..., 10 ; les tâches 1 à 4 peuvent elles-même être réalisées en parallèle, etc.



Dates de début au plus tôt : on note $t(i)$ la date de début au plus tôt de la tâche i , sachant que le projet commence à la date $t(0) = 0$. Pour le projet de l'exemple ci-dessus : $t(0) = 0, t(1) = 0, \dots, t(5) = 0, t(6) = 8, \dots$

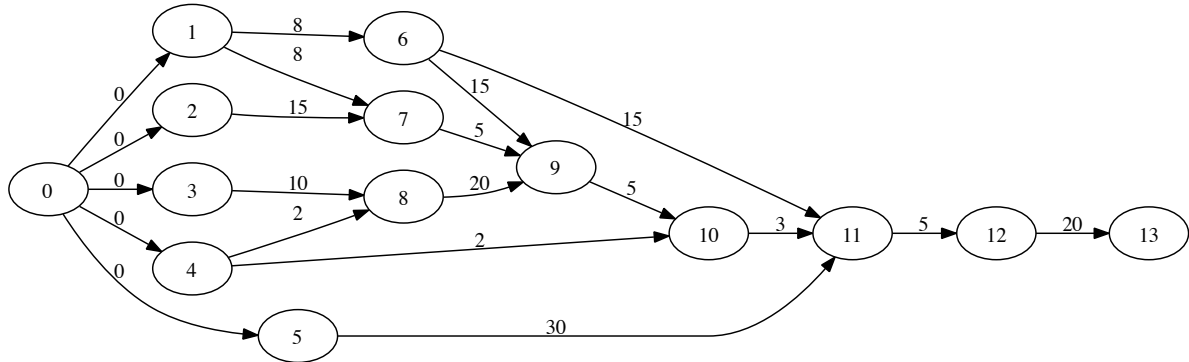
La tâche « fin du projet » étant de durée nulle, sa date de début au plus tôt, $t(n-1)$, est le temps minimum nécessaire à la réalisation du projet : la tâche « fin du projet » commencera et sera instantanément réalisée à la date $t(n-1)$.

Tâches et chemins critiques : une tâche est « critique » si une augmentation de sa durée entraîne une augmentation de la durée de réalisation du projet. À titre d'illustration, les tâches 9 et 12 sont critiques, les tâches 5 et 6 ne le sont pas. Dans le suivi d'un projet les tâches critiques doivent être surveillées avec la plus grande attention car le moindre retard dans leur réalisation entraîne un allongement de la durée totale du projet. Un chemin critique est une suite de tâches critiques, $c(0), \dots, c(k-1), c(k), \dots, c(q-1)$, telle que $c(0) = 0, c(q-1) = n-1$, et pour tout k , $c(k-1) \in pred(c(k))$.

Représentation : les données du problème sont représentées par la table des arcs du graphe du projet et par le vecteur $d[0..n-1]$ des durées des tâches (vecteur de terme général $d[i] = d_i$.)

On veut calculer la durée minimum du projet, c'est-à-dire la date $t(n - 1)$ de début au plus tôt de la tâche « fin du projet », et afficher les tâches d'un chemin critique.

1. Indiquer sur chaque tâche de l'exemple ci-dessous sa date de début au plus tôt et marquer d'une étoile les tâches d'un chemin critique. Sur cet exemple il existe plusieurs chemins critiques, n'en marquer qu'un seul ;



2. donner les équations de récurrence des dates de début au plus tôt (base et cas général) ;
3. donner un programme calculant un tableau $T[0..n - 1]$ de terme général $T[i] = t(i)$ et un tableau $A[0..n - 1]$ de terme général $A[i] = \text{arg}(t(i))$, c'est-à-dire un prédécesseur de i pour lequel la date $t(i)$ de début au plus tôt a été obtenue ;
4. donner la complexité du programme en fonction du nombre de tâches (n) et du nombre d'arcs (m) ;
5. donner un programme d'affichage d'un chemin critique ;
6. exécuter vos programmes sur le projet donné en exemple.

Ces programmes feront l'objet du TP2 (jeudi 28 ou vendredi 29 mai selon les groupes.) Les TP sont à faire en binôme. Ils sont à rendre au début de la dernière séance de TP (9, 10 ou 12 juin.)