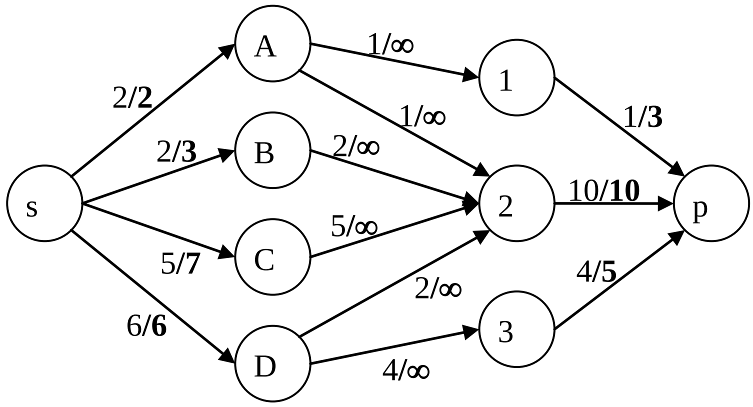
Final RO03 – Printemps 2012 - Jacques Carlier

EXERCICE 1 (4 points) : FLOT MAXIMAL

Dans le graphe ci-dessous, on a rapporté les flots et les capacités de chacun des arcs sous la forme flot/**capacité** (notons que les arcs du milieu sont de capacité infinie).



Calculer un flot maximal de *s* à *p* sur le graphe. On rapportera le flot maximal (en précisant la (ou les) chaîne(s) améliorante(s) ayant permis de construire ce flot) et la coupe minimale obtenue à la fin de l'algorithme.

EXERCICE 2 (4 points) .

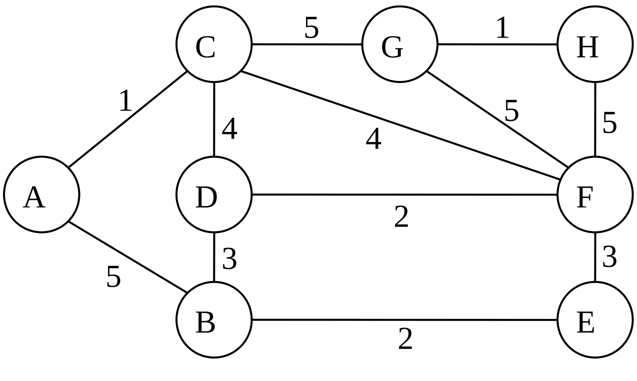
On considère le problème d'ordonnancement donné par le tableau ci-dessous :

Tâches	a	b	c	d	e	f
Durés	4	8	10	4	1	11
Tâches préalables	_	_	a	a et b	c	_

1) Dessiner le graphe potentiel-tâches. On notera 0 et n+1, les tâches fictives de début et de fin.
2) Calculer sur le graphe potentiel-tâches, les dates au plus tôt ainsi que les dates au plus tard. Donner le chemin critique de ce graphe.
3) Dessiner un graphe PERT, avec un seul arc fictif représentant le problème.
4) Introduisez sur le graphe potentiel-tâches une date de disponibilité de 5 pour la tâche b. Quelle est la nouvelle date de fin au plus tôt du projet ?

PROBLEME (12 points)

On a n objets à classer et pour certaines paires d'objets [i, j] un indice de dissimilarité $a_{[i,j]} \in \{0,1, \ldots, D\}$ avec $a_{[i,i]} = 0$ pour tout i. À ces données, on associe le graphe G non orienté de n sommets X={1, 2,…, n} (représentant les objets) et où l'arête (i, j) est valuée par $a_{[i,j]}$. Dans la suite, on supposera G connexe à m arêtes.



le graphe G

(les boucles de valuation 0 n'ont pas été représentées)

Pour tout $\alpha \in \{0,1, \ldots, D\}$, on définit le graphe $G_\alpha = (X, E_\alpha)$ avec $E_\alpha = \{ (i,j) / \ a_{[i,j]} \leq \alpha \}$.

Définition : **Les classes de classification de niveau α sont les composantes connexes du graphe G_α.**

À chaque α correspond alors une partition des sommets, où chaque sous-ensemble de sommets de la partition correspond à une composante connexe. On rappelle que la partition d'un ensemble X est un ensemble $\{X_1, ..., X_k\}$ de sous-ensembles de sommets non vides deux à deux disjoints et dont la réunion constitue l'ensemble de départ.

Les différents niveaux de classification obtenus en faisant varier α de 0 à D définissent alors un ensemble de partitions des n objets.

Question 1) Tracer G_0, G_1, G_2, G_3, G_4, G_5 pour l'exemple ci-dessus et déterminer les partitions de niveau α=0,1,2,3,4,5 pour ce graphe

Question 2) Quelle est la complexité du calcul de l'ensemble de ces partitions en fonction de n, de m et de D ? On pourra supposer que l'on utilise le meilleur algorithme de recherche de composantes connexes vu en TD. On expliquera brièvement le calcul.

Question 3) Les différents niveaux de classification obtenus en faisant varier α de 0 à m définissent donc un ensemble de partitions des n objets. On associe à un tel graphe une arborescence T, dont les sommets sont les classes de cette partition. Dessiner l'arborescence T de l'exemple.

Question 4) Appliquer l'algorithme de Kruskal au graphe G.

Question 5) Appuyer vous sur l'algorithme de Kruskal pour obtenir toutes les partitions et l'arborescence de classification. Expliquer. Quelle est la complexité de cette deuxième méthode si la complexité de Kruskal est O(m log n) ? Que gagne-t-on par rapport à la méthode de la question 2 ?

A toute chaîne μ = [x1,x2] [x2,x3] …[xk-1,xk] de G, on associe la mesure suivante :

$d(\mu) = \max \{ a_{[x1,x2]}, a_{[x2,x3]}, \ldots, a_{[xk-1,xk]} \}$, c'est-à-dire le maximum des valuations de la chaîne.

Soit L_{ij}, l'ensemble des chaînes entre i et j, d^*_{ij} est par définition la distance minimale de i à j sur l'ensemble des chaînes de L_{ij}.

Question 6) Montrer que :

- $\forall i \in X, d^*_{ii} = 0$
- $\forall i,j \in X, d^*_{ij} = d^*_{ji}$
- $\forall i,j,h \in X \quad d^*_{ij} \leq \max\{ d^*_{ih}, d^*_{hj} \}$

Question 7) Ecrire en vous inspirant de la méthode matricielle un algorithme qui calcule les valeurs d^*_{ij}. Quelle est la complexité de cet algorithme ? Justifier brièvement cet algorithme.

Remarque : les questions 4 , 6 et 7 sont indépendantes des autres questions du problème.

<u>Rappel sur l'algorithme de KRUSKAL :</u>

L'algorithme de KRUSKAL a pour objet de calculer un arbre couvrant de somme de valuations minimale. Il consiste à prendre les arêtes dans un ordre de coût croissant et à ne retenir que les arêtes qui ne créent pas de cycle quand on les introduit dans l'arbre en cours de construction :

Numéroter les arêtes du graphe $\{u_1, ..., u_m\}$ dans un ordre de valuations croissantes ;

$T \leftarrow (X, \emptyset)$;

pour $k \leftarrow 1$ à m faire {

Si $T+\{u_k\}$ ne ce contient pas de cycle alors $T \leftarrow T + \{u_k\}$;

}