Université de Strasbourg — **UFR de Mathématiques et Informatique**
L2 Informatique S4 — **2013-2014, semestre de printemps**

Graphes

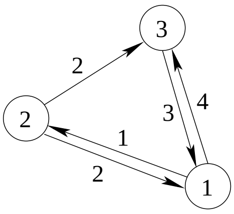
Contrôle Continu n°3 — Convoqué

Durée : 1 heure
Responsable : Prof. Christian RONSE
Tous documents en papier autorisés mais non partagés
Calculettes inutiles
Téléphones et appareils électroniques éteints et rangés dans un sac fermé

Justifiez soigneusement vos réponses !

NB. *Les graphes seront toujours supposés* **finis**, *c.-à-d. avec un nombre fini de sommets et un nombre fini d'arêtes.*

(1) Distances pondérées entre toutes les paires de sommets.
On a un graphe orienté à 3 sommets s_1, s_2, s_3 et des arcs (s_i, s_j), avec des poids $p(i,j)$ comme suit :



$p(1,2) = 1,\ \ p(1,3) = 4,\ \ p(2,1) = 2,\ \ p(2,3) = 2,\ \ p(3,1) = 3,$
mais pas d'arc de s_3 vers s_2, et aucune boucle, voir ci-contre.
Calculer la matrice de distances pondérées $d(i,j)$ entre toutes les paires (s_i, s_j) de sommets par l'algorithme de Floyd-Warshall.

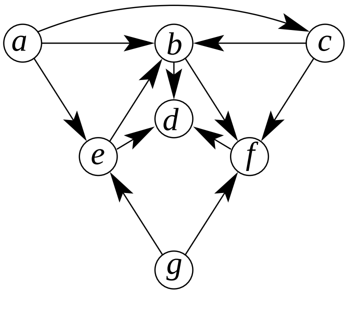
Rappel : Pour un graphe orienté à n sommets, l'algorithme de Floyd-Warshall donne $d(i,j) = D^{(n)}(i,j)$, où $D^{(k)}(i,j)$ est calculé par récurrence sur k comme suit : pour $k = 0$ on a

$$D^{(0)}(i,i) = 0 \quad \text{et pour } i \neq j,\ D^{(0)}(i,j) = \begin{cases} p(i,j) & \text{s'il y a un arc de poids } p(i,j) \text{ de } s_i \text{ vers } s_j, \\ \infty & \text{s'il n'y a pas d'arc de } s_i \text{ vers } s_j \text{ ;} \end{cases}$$

puis pour $k > 0$ on a

$$D^{(k)}(i,j) = \min\Big(D^{(k-1)}(i,j), D^{(k-1)}(i,k) + D^{(k-1)}(k,j)\Big) \ .$$

(2) Tri topologique.



Le graphe orienté ci-contre n'a pas de circuit. Effectuer un tri topologique de ce graphe à l'aide du parcours en profondeur.

(3) Arbre couvrant minimum.
Un graphe simple non orienté connexe G a m arêtes $a_1, \ldots, a_m$, et chacune a un poids strictement positif : $p(a_1), \ldots, p(a_m)$, où $p(a_i) > 0$ pour $i = 1, \ldots, m$. On peut construire un arbre couvrant minimum par un des 3 algorithmes vus en cours : ceux de Kruskal et Prim (tous deux par ajouts successifs d'arêtes), et celui par retraits successifs d'arêtes.

Supposons qu'on modifie les poids en leur appliquant une fonction strictement croissante sur les réels positifs, c.-à-d. on a de nouveaux poids $q(a_1), \ldots, q(a_m)$, toujours strictement positifs, $q(a_i) > 0$, avec les contraintes suivantes pour tous les indices i, j :

si $p(a_i) = p(a_j)$, alors $q(a_i) = q(a_j)$;

si $p(a_i) < p(a_j)$, alors $q(a_i) < q(a_j)$.

Par exemple, on peut augmenter les poids d'une constante positive, les doubler, les mettre au carré, etc.

Question : Avec ces nouveaux poids, le déroulement de ces 3 algorithmes d'arbre couvrant minimum sera-t-il modifié ? Le résultat sera-t-il différent ?