

## - Examen du 9/1/2007. -

- Durée 2h. Aucun document autorisé. -

- Une large part de la notation prendra en compte la clarté de la rédaction et la rigueur des justifications. -

## - Exercice 1 - Algorithme de Floyd-Warshall.

- Ecrire l'algorithme de Floyd-Warshall prenant en entrée la matrice  $W$  des poids des arcs d'un graphe orienté  $D$  et retournant la matrice des poids minimum des chemins orientés entre tous les couples de sommets.
- Appliquer l'algorithme aux deux matrices suivantes. Que peut-on déduire du résultat ?

$$W_1 = \begin{pmatrix} 0 & 2 & -1 & +\infty \\ +\infty & 0 & 1 & -2 \\ +\infty & +\infty & 0 & 1 \\ 1 & +\infty & +\infty & 0 \end{pmatrix} \quad W_2 = \begin{pmatrix} 0 & 2 & -3 & +\infty \\ +\infty & 0 & 1 & -2 \\ +\infty & +\infty & 0 & 1 \\ 1 & +\infty & +\infty & 0 \end{pmatrix}$$

- Modifier l'algorithme de Floyd-Warshall afin de calculer les chemins orientés de poids minimum et non plus seulement les poids des chemins.
- Que retourne votre algorithme modifié sur la matrice  $W_1$  ?

## - Exercice 2 - Tri topologique.

On se donne le graphe orienté  $D$  sur l'ensemble de sommets  $V = \{1, \dots, 7\}$  dont la matrice d'adjacence  $M$  est donnée ci-dessous :

$$M = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

On rappelle qu'un *tri topologique* d'un graphe orienté  $D$  est une énumération  $x_1, \dots, x_n$  des sommets de  $D$  telle que tout arc  $x_i x_j$  de  $D$  vérifie  $i < j$ .

- Calculer un tri topologique de  $D$  en précisant votre méthode.
- Combien y a-t-il d'autres tris topologiques de  $D$  ? Justifier.
- Combien d'arcs au plus possède un graphe orienté sur  $n$  sommets admettant un tri topologique ?
- Montrer qu'un graphe orienté admet un tri topologique si et seulement si pour tout couple de sommets distincts  $(u, v)$  il n'existe pas de chemin orienté de  $u$  à  $v$  ou il n'existe pas de chemin orienté de  $v$  à  $u$ .

- **Exercice 3 - Circuit de longueur minimale** - Un graphe orienté  $D = (V, A)$  est *fortement connexe* si pour tout couple de sommets distincts  $(u, v)$ , il existe un chemin orienté de  $u$  à  $v$ . Dans tout cet exercice, on considère que les arcs de  $D$  possèdent une longueur positive donnée par une fonction  $l : A \rightarrow \mathbb{R}^+$ .

- Utiliser l'algorithme de Dijkstra pour trouver une arborescence des plus courts chemins depuis le sommet  $s$  dans le graphe pondéré de la Figure 1. Vous détaillerez les 3 premières étapes de votre calcul.

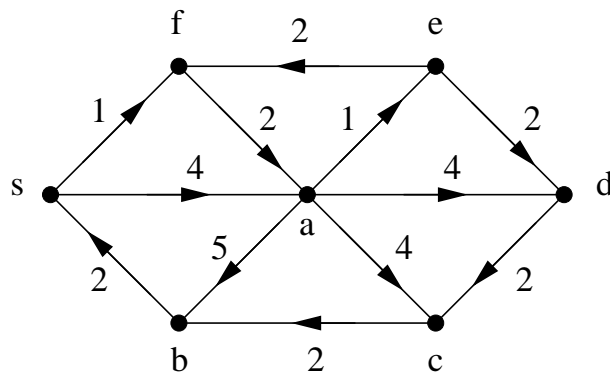


FIG. 1 – Le graphe orienté pondéré

- Montrer que dans un graphe orienté fortement connexe tout arc est inclus dans un circuit.
- Donner un circuit de longueur totale minimum pour le graphe de la Figure 1. Justifier votre réponse.
- Si  $C$  est un circuit de longueur totale minimum passant par un arc  $uv$ , justifier que  $C - uv$  est un plus court chemin de  $v$  à  $u$ .
- On rappelle que le temps d'exécution d'une implémentation standard de l'algorithme de Dijkstra est en  $O(m \cdot \log n)$ , où  $n$  est le nombre de sommets du graphe d'entrée et  $m$  son nombre d'arcs. On s'intéresse au problème de trouver un circuit de longueur totale minimum dans un graphe orienté fortement connexe et dont les arcs ont une longueur positive. Donner le principe d'un algorithme résolvant ce problème avec un temps d'exécution de  $O(n \cdot m \cdot \log n)$ .

Barème approximatif :

Exercice 1 : 2+2+2+1

Exercice 2 : 2+1+1+2

Exercice 3 : 2+1+1+1+2