

Devoir Surveillé du 13 novembre 2017
durée 1h30

Dans tout le sujet, on conviendra que, dans les différentes structures de données modélisant les graphes, les sommets sont rangés dans l'ordre croissant de leur numéro.

Exercice 1

Soit G un graphe orienté. Un sommet de G est une *source*, s'il n'a pas de voisin entrant. Un sommet de G est un *puits*, s'il n'a pas de voisin sortant.

1. Identifier toutes les sources et tous les puits du graphe G_1 de la figure 1.
2. Donner un exemple d'un graphe orienté qui n'a ni source ni puits.
3. Pour chacune des trois représentations (matrice d'adjacence, matrice d'incidence, listes d'adjacence), proposer un algorithme $\text{Puits}(G, s)$ qui retourne **True** si un sommet donné s d'un graphe orienté donné G est un puits, et qui retourne **False** sinon ; et un algorithme $\text{Source}(G, s)$ qui retourne **True** si un sommet donné s d'un graphe orienté donné G est une source, et qui retourne **False** sinon.

Étudier la complexité des six algorithmes, en fonction du nombre de sommets n , du nombre d'arcs m et du degré sortant maximum d .

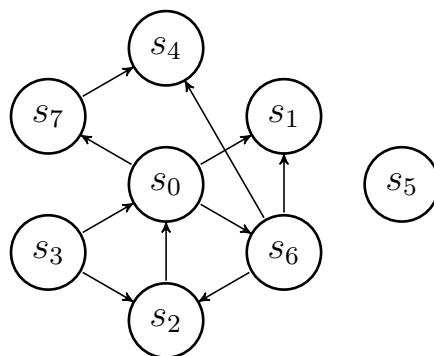


FIGURE 1 – Le graphe G_1

Exercice 2

Pour chacun des deux graphes G_2 et G_3 de la figure 2, décider s'il existe un tri topologique. Si c'est le cas, donner le tri topologique obtenu en appliquant l'algorithme du cours, sinon, justifier.

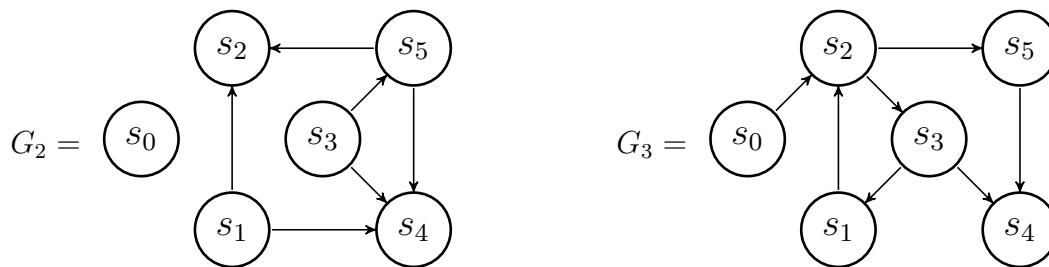


FIGURE 2 – Les graphes G_2 et G_3

Exercice 3

Soit G un graphe orienté fini tel que le degré sortant de chaque sommet de G est égal à 1. (Il n'y a pas de contraintes sur les degrés entrants.) Considérons les types d'arcs lors d'un parcours en profondeur de G .

1. Montrer que le parcours en profondeur du graphe G ne contient aucun arc avant.
2. Montrer que dans le parcours en profondeur, tout sommet de G a au plus un fils dans l'arborescence (c'est à dire est l'origine d'au plus un arc de liaison).
3. Montrer que toute composante de l'arborescence des arcs de liaison est formée par un et une seul chemin (éventuellement trivial), dont le dernier sommet est incident à un arc sortant transverse ou de retour.
4. Soit G_0 le graphe non-orienté obtenu à partir de G en oubliant l'orientation des arcs. Montrer que toute composante connexe de G_0 possède un et un seul cycle.

RAPPEL :

<pre> 0 PP(G) 1 pour chaque sommet u de V faire 2 couleur[u] <- BLANC 3 pere[u] <- nil 4 temps <- 0 5 pour chaque sommet u de V faire 6 si couleur[u] = BLANC 7 alors Visiter_PP(u) </pre>	<pre> 0 Visiter_PP(u) 1 couleur[u] <- GRIS 2 d[u] <- temps <- temps + 1 3 pour chaque v de Adj[u] faire 4 si couleur[v] = BLANC 5 alors pere[v] <- u 6 Visiter_PP(v) 7 couleur[u] <- NOIR 8 f[u] <- temps <- temps + 1 </pre>
---	---