Examen d'algorithmique

Mardi 19 juin 2018 15h30 – 17h30 / Aucun document autorisé

Mode d'emploi : Le barème est donné à titre indicatif. **La qualité de la rédaction sera très fortement prise en compte pour la note.** On peut toujours supposer une question résolue et passer à la suite. L'annexe contient plusieurs algorithmes vus en cours.

Exercice 1 : Recherche des composantes fortement connexes – (4 points)

On rappelle l'algorithme vu en cours pour la recherche des composantes fortement connexes d'un graphe orienté $G = (S, A)$:

1. Appeler PPC(G) pour obtenir des dates f pour chaque sommet ;
2. Construire G^t (le graphe transposé de G) ;
3. Appeler PPC(G^t) en énumérant les sommets par date f[–] décroissante ;
4. Retourner les arbres de parcours obtenus par PPC(G^t) : ce sont les CFC.

Appliquer l'algorithme au graphe de la figure (1) en détaillant chaque étape : on donnera le résultat du premier parcours sous la forme des dates de début et de fin de traitement de chaque sommet (coloriage en gris et en noir), le graphe transposé et le résultat du parcours en profondeur du graphe transposé avec les dates et l'arborescence de ce parcours. On en déduira les composantes fortement connexes.

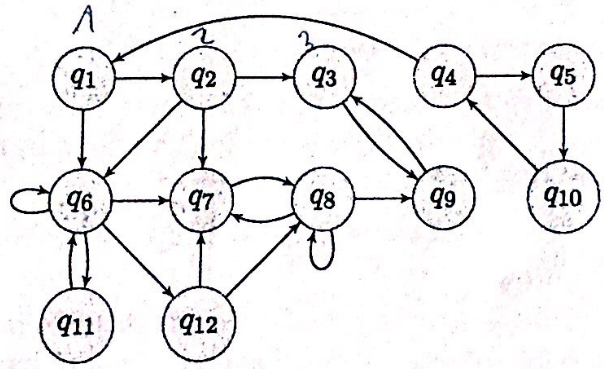

FIGURE 1 – Graphe de l'exercice 1

Exercice 2 : Autour de l'algorithme de Dijkstra – (5 points)

1. Rappeler à quelle condition, si un graphe orienté valué $G = (S, A, w)$ contient des arcs de poids négatif, il existe un plus court chemin entre deux sommets x et y de S.
2. Montrer qu'en présence d'arcs de poids négatif, l'algorithme de Dijkstra peut échouer à trouver un plus court chemin (même si il en existe bel et bien).
3. Soit $G = (S, A, w)$ un graphe orienté valué avec des poids négatifs. On note p_G le plus petit poids apparaissant sur un arc de G. On définit le graphe G_p par $G_p = (S, A, w_p)$ avec $w_p(x, y) = w(x, y) - p_G$.

   (a) Est-ce que G_p contient des arcs de poids strictement négatif ?

(b) Soit ρ un chemin dans G (et dans G_p).
   i. Quel est le lien entre $w(\rho)$ et $w_p(\rho)$?
   ii. Si ρ est un plus court chemin dans G, est-ce que ρ est un plus court chemin dans G_p ?

(c) Peut-on appliquer l'algorithme de Dijkstra pour trouver les plus courts chemins dans G_p et en déduire ceux de G ?

Exercice 3 : Recherche d'arbres couvrants minimaux – (3 points)

1. Rappeler ce qu'est un arbre couvrant minimal d'un graphe non-orienté valué $G = (S, A, w)$. Donner deux ACM du graphe de la figure ci-dessous.

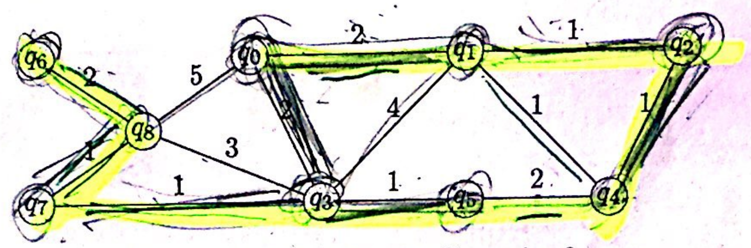

FIGURE 2 – Graphe pour l'exercice 2.

2. Expliquer comment procède un des algorithmes pour la recherche d'ACM vus en cours.
3. Appliquer l'algorithme choisi au graphe ci-dessus (on indiquera quelle arête est ajoutée au résultat à chaque étape).

Exercice 4 : Autour des parcours – (5 points)

1. Dans cette question, on veut construire deux graphes orientés pour lesquels un parcours en profondeur peut renvoyer certaines dates d et f données ci-dessous. **Les deux graphes proposés devront contenir au moins un circuit.**
   (a) Graphe n°1 : un parcours en profondeur doit pouvoir donner les dates du tableau n°1 ci-dessous.
   (b) Graphe n°2 : on veut un graphe qui peut, selon la manière dont on exécute l'algorithme, avoir les dates du tableau n°2 et celles du tableau n°3. On impose en plus que chaque sommet peut atteindre tous les autres.

Tableau n°1 :

	q_1	q_2	q_3	q_4
d	3	1	2	6
f	4	8	5	7

Tableau n°2

	q_1	q_2	q_3	q_4
d	1	2	3	4
f	8	7	6	5

Tableau n°3 :

	q_1	q_2	q_3	q_4
d	2	3	1	6
f	5	4	8	7

2. Ecrire un algorithme `CycleAtteignable`(G, x) qui étant donnés un graphe G et un sommet x retourne *Vrai* si et seulement si il est possible d'atteindre un cycle depuis le sommet x.

3. Etant donnés la table Π renvoyée par un l'algorithme de parcours en profondeur sur un graphe $G = (S, A)$, et deux sommets x et y de S, écrire un algorithme `TestRacine`(Π, x, y) qui renvoie `Vrai` si et seulement x et y ont la même racine dans la forêt de parcours définie par Π.

Exercice 5 : Flots (3 points)

On considère le réseau de transport R de la figure 3 :

1. Appliquer l'algorithme de Ford-Fulkerson pour trouver un flot maximal. On donnera les flots intermédiaires et les graphes des augmentations obtenus par l'algorithme (une feuille est préremplie, voir pages 5 et 6 du sujet). Attention : **on demande à ce que votre application de l'algorithme utilise au moins un arc arrière.**
2. En déduire une coupe de capacité minimale du réseau R.

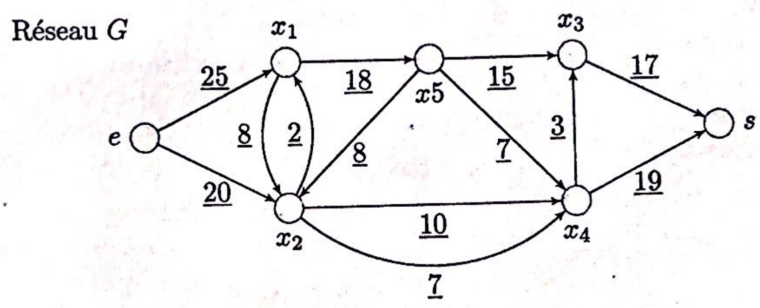

FIGURE 3 – Le réseau de transport R pour l'exercice 5.