- Examen de 2ème session -

- Durée : 2 heures - Aucun document n'est autorisé -
Juin 2016

Le barème est indicatif. Les exercices peuvent être traités dans un ordre quelconque.
Toutes les réponses doivent être claires et justifiées.

- Exercice 1 - Parcours en profondeur - *(4 points)*

Pour un entier $p \geq 2$, le graphe CD_p a pour sommets $\{1, 2, \ldots, 2p\}$ et pour arêtes $\{\{i, i+1\} \ : \ i = 1, \ldots, 2p-1\} \cup \{\{2p, 1\}\} \cup \{\{i, i+p\} \ : \ i = 1, \ldots, p\}$. On suppose que le graphe CD_p est codé par liste de voisins. Pour $i \in \{1, \ldots, 2p\}$, on notera $L(i)$ la liste des voisins du sommet i. On souhaite effectuer un parcours en profondeur de CD_p avec la règle $(*)$ suivante : en cas de choix pour le successeur d'un sommet i, on choisira un sommet non déjà visité de $L(i)$ ayant la même parité que i si un tel sommet existe, sinon on choisira un sommet non déjà visité de $L(i)$ de numéro de sommet le plus grand possible.

a. Représenter le graphe CD_4.

b. Effectuer un parcours en profondeur de CD_4 en respectant la règle $(*)$ et en partant du sommet 1. Indiquer pour chaque sommet, son père, sa date de première visite (`début`) et sa date de dernière visite (`fin`).

c. Pour $p \geq 2$ quelconque, donner la liste des sommets de CD_p par date de `début` croissante après un parcours en profondeur respectant $(*)$ et de racine le sommet 1.

d. L'arbre de parcours obtenu possède une propriété spéciale, quelle est-elle ? Comment nomme-t-on une telle structure ?

- Exercice 2 - Parcours en largeur, détection de cycle - *(6 points)*

On s'interesse à la détection de cycle dans un graphe.

a. On se donne l'algorithme CYCLE suivant, fonctionnant sur un graphe $G = (V, E)$:

> **Algorithme :** CYCLE
> **pour tous les** $x \in V$ **faire**
>   **pour tous les** $y \in V$ *avec* $y \neq x$ **faire**
>   **si** *(x et y sont voisins et ont un voisin en commun)* ***ou*** *(x et y ont deux voisins distincts en commun)* **alors**
>   **retourner** `vrai` ;
> **retourner** `faux` ;

Appliquer cet algorithme sur le graphe CD_4 de l'exercice précédent.

b. L'algorithme CYCLE appliqué à un graphe G quelconque renvoie-t-il `vrai` si, et seulement si, G contient un cycle ? Justifier votre réponse (c-à-d donner une preuve ou un contre exemple).

c. Effectuer un parcours en largeur du graphe CD_4 de l'exercice précédent en partant du sommet 1 comme racine. Indiquer pour chaque sommet, son père ainsi que son niveau.

d. Démontrer qu'un graphe G connexe est sans cycle si, et seulement si, pour toute arête xy de G, on a $x = pere(y)$ ou $y = pere(x)$, où *pere* est la fonction *pere* retournée par un parcours en largeur de G.

e. En déduire un algorithme qui, pour un graphe G connexe fourni en entrée, détecte si G contient un cycle et le cas échéant écrit les sommets du cycle détecté (on pourra faire appel à un algorithme $\texttt{PL}(G, \mathit{pere}, \mathit{niveau})$ qui effectue un parcours en profondeur de G et remplit les tableaux *pere* et *niveau* correspondants). Préciser la complexité de votre algorithme.

f. Écrire un algorithme qui prend en entrée un graphe G connexe et un sommet x de G et qui retourne un cycle de G de longueur impaire contenant x ou une information signifiant qu'un tel cycle n'existe pas (vous pourrez encore faire appel à $\texttt{PL}(G, \mathit{pere}, \mathit{niveau})$). Préciser la complexité de votre algorithme et prouver sa validité.

- Exercice 3 - Moins chers chemins - *(4 points)*

1. *Question de cours.* Rappeler le déroulement de l'algorithme de Bellman-Ford (permettant de calculer des plus courts chemins depuis une racine dans un graphe orienté dont les arcs sont valués positivement ou négativement), ainsi que sa complexité. Quelle condition faut-il imposer sur le graphe d'entrée pour assurer le fonctionnement de l'algorithme ?
2. Un transporteur doit livrer des marchandises dans différentes villes depuis la ville origine s. Chaque trajet possible d'une ville x vers une ville y a un coût associé $c(xy)$. Le transporteur profite toutefois du trajet de x à y pour ré-équilibrer ses entrepots en transportant des marchandises pour son compte, générant ainsi un gain $g(xy)$. Les coûts et gains des trajets possibles entre les villes s, a, b, c, d et e sont donnés ci-dessous (en k€). Le trajet de x à y est noté xy.

trajet xy	sa	sb	bc	bd	ad	ca	ce	de
coût $c(xy)$	5	6	4	6	4	2	3	7
gain $g(xy)$	0	0	3	4	5	6	5	1

   On souhaite calculer pour chaque ville l'itinéraire le moins cher depuis la ville s. Modéliser le problème.
3. Résoudre le problème.

- Exercice 4 - Hypercube - *(6 points)*

L'*hypercube* H_n de dimension n est le graphe dont les sommets sont $\{0,1\}^n$ (un tel sommet est un mot de n lettres sur l'alphabet $\{0,1\}$, on le notera $x_1 \dots x_n$ ou $y_1 \dots y_n$). Deux sommets $x_1 \dots x_n$ et $y_1 \dots y_n$ sont reliés dans H_n si il existe $i \in \{1, \dots, n\}$ tel qu'on aît $x_i \neq y_i$ et que pour tout $j \neq i$ on aît $x_j = y_j$. Une telle arête sera dite de *dimension* i.

a. Montrer qu'il existe un chemin de $0^n = 00\dots0$ à $1^n = 11\dots1$ dans H_n.

b. Plus généralement, montrer que H_n est connexe.

c. On munit chaque arête de dimension i d'un poids égal à i. À l'aide de l'algorithme de Kruskal, calculer un arbre couvrant de poids minimum de H_3 muni du poids précédent.

d. Un *code de Gray de dimension* n est une suite de mots de $\{0,1\}^n$ formant un chemin hamiltonien de H_n. Proposer un code de Gray de dimension 3.

e. Montrer que pour tout $n \geq 1$, il existe un code de Gray de dimension n (on pourra procéder par récurrence sur n).

f. *Hors barême.* Proposer un poids sur les arêtes de H_n afin qu'un arbre couvrant de H_n muni de ce poids corresponde à un code de Gray.