- Examen -

- Durée : 2 heure - Aucun document n'est autorisé -
Janvier 2015

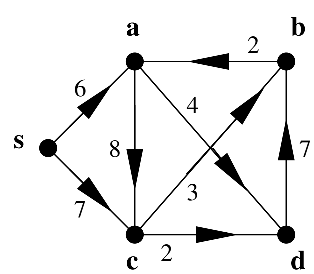
Le barême est indicatif. Les exercices peuvent être traités dans un ordre quelconque.
Toutes les réponses doivent être claires et justifiées.

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

1. En suivant les indications qui suivent, effectuer un parcours en profondeur dans le graphe de la grille 5×5 (ses sommets sont $\{v_{i,j} \ : \ 1 \leq i \leq 5 \ \text{ et } 1 \leq j \leq 5\}$ et ses arêtes $\{v_{i,j}v_{i',j'} \ : \ (i = i' \text{ et } |j - j'| = 1) \text{ ou } (j = j' \text{ et } |i - i'| = 1)\}$). On suppose que la liste des voisins de chaque sommet v est lue dans l'ordre suivant (sous réserve d'existence du voisin en question) : le sommet en haut de v, puis celui à droite de v, puis celui en bas de v, et enfin celui à gauche de v. La racine est le sommet en bas à gauche. Seul l'arbre de parcours sera représenté.
2. Même question lorsque les voisins sont lus dans l'ordre haut, bas, droite, gauche.
3. Approximer le nombre total de lectures de voisins NTL_n pour chacun de ces deux parcours sur la grille $n \times n$ *au moment ou le dernier sommet de la grille est découvert.* Votre réponse sera de la forme $NTL_n \equiv c.n^2$ où c est une constante.

- Exercice 2 - Plus courts chemins - *(7 points)*

On considère le graphe orienté et pondéré ci-dessous :



1. Utiliser l'algorithme de Dijsktra sur le graphe ci-dessus afin de déterminer une arborescence des plus courts chemins issue de s. Indiquer rapidement les diverses étapes du déroulement de votre algorithme.
2. *Question de cours.* Rappeler l'algorithme de Bellman-Ford, ainsi que sa complexité.
3. Les longueurs des arcs ad, ba et cb sont modifiées et deviennent respectivement -4, -2 et -3. Utiliser maintenant l'algorithme de Bellman-Ford afin de déterminer une arborescence des plus courts chemins issue de s. Prenez l'ordre lexicographique comme ordre de traitement des arcs.
4. Que se passe-t-il si on souhaite utiliser l'algorithme de Bellman-Ford sur un graphe possédant un circuit de longueur totale négative ? Modifier cet algorithme afin qu'il détecte et écrive, le cas échéant, un tel circuit.

- Exercice 3 - (Re) plus courts chemins - *(3 points)*

Soit G un graphe connexe non orienté, sans poids sur les arêtes et dont l'ensemble des sommets est $\{1,\ldots,n\}$. Le but de cet exercice est de stocker un ensemble de plus courts chemins entre tous les couples de sommets sous la forme d'un tableau $C = (c_{ij})$ de taille $n \times n$. Un plus court chemin entre deux sommets k et l se calcule alors avec l'algorithme suivant :

PLUS-COURT-CHEMIN(k, l)
1 Écrire k
2 Tant que $c_{k,l} \neq \emptyset$ faire
3 Écrire $c_{k,l}$
4 $k \longleftarrow c_{k,l}$

1. Dérouler l'algorithme précédent sur le tableau suivant :

$$C = \begin{pmatrix} \emptyset & 2 & 2 & 4 \\ 1 & \emptyset & 3 & 1 \\ 2 & 2 & \emptyset & 4 \\ 1 & 1 & 3 & \emptyset \end{pmatrix}$$

afin de calculer un plus court chemin entre 1 et 3, puis entre 3 et 4. Quel est le graphe qui correspond à ce tableau C ?

2. Calculer un tableau C pour le graphe dont les sommets sont $\{1, 2, 3, 4, 5\}$ et l'ensemble des arêtes $\{12, 23, 34, 45, 15, 25\}$. Expliquer brièvement la méthode utilisée.

- Exercice 4 - Graphe sans triangle - *(6 points)*

Un *triangle* dans un graphe G est un cycle de longueur 3, c'est-à-dire un sous-graphe de G sur trois sommets $\{x, y, z\}$ tel que les trois arêtes xy, yz et xz existent dans G.
Le but de l'exercice est de proposer des algorithmes pour reconnaître un graphe sans triangle, puis de montrer qu'un graphe sans triangle avec n sommets et m arêtes vérifie $m \leq n^2/4$.

1. Dessiner un graphe sans triangle.
2. Si ce n'était déjà pas le cas, dessiner un graphe sans triangle qui ne soit pas biparti.
3. En supposant que les graphes d'entrée soient codés sous forme de matrice d'adjacence, proposer un algorithme en $O(n^3)$ qui décide si un graphe est sans triangle ou non.
4. On peut remarquer que si les deux extémités d'une arête d'un graphe G ont un voisin en commun alors G contient un triangle. Déduire de cette observation :
   (a) un algorithme en $O(nm)$ qui décide si un graphe est sans triangle ou non.
   (b) si G est sans triangle, alors pour toute arête xy de G, on a $d(x) + d(y) \leq n$ (où $d(x)$ désigne le degré d'un sommet x).
5. Montrer que $\sum_{uv\in E(G)}(d(u) + d(v)) = \sum_{u\in V(G)} d^2(u)$ et déduire de la question *4(b)* que si G est sans triangle alors $\sum_{u\in V(G)} d^2(u) \leq nm$.
6. En utilisant l'inégalité (de Cauchy-Schwartz) : $(\sum_{i=1}^{n} a_i b_i)^2 \leq (\sum_{i=1}^{n} a_i^2)(\sum_{i=1}^{n} b_i^2)$ et la question (5) conclure l'exercice.
7. Pour n pair quelconque, proposer un graphe sans triangle à n sommets et m arêtes vérifiant $m = n^2/4$.