Devoir Surveillé Algorithmie Avancée

L3 MPCI

11 octobre 2024 - Durée: 2h

Lorsque l'on vous demande d'écrire de décrire ou de donner un algorithme cela signifiera toujours en donner un pseudo-code, justifier de son exactitude et de sa complexité

On rappelle qu'aucun document ni équipement électrique ou électronique n'est autorisé.

Les exercices :
- sont au nombre de 4 ;
- sont indépendants ;
- ont un début plus facile que la fin.

Exercice 1 – Débit de réseau

Dans un réseau de communication, on appelle *débit* la quantité d'information que le réseau garantit de pouvoir faire passer entre deux sommets. Dans cet exercice, le réseau est modélisé par un graphe $G = (X, E)$ connexe. Chaque arête est munie d'une bande passante (qui ici sera appelée poids), $v : E \to \mathbb{R}^+$, qui limite la quantité d'information qu'elle peut véhiculer. Le but de l'exercice est de mettre au point des algorithmes permettant de calculer le débit. Le graphe de la figure 1 va servir d'exemple.

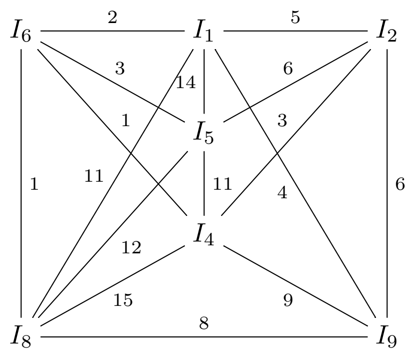

Figure 1 – Un réseau de transport

1.1

Question 1.1.1 Montrer que si $\mathcal{C}_{xy}$ est l'ensemble des chemins entre x et y alors le débit entre x et y s'écrit :

$$D(x, y) = \max(\{\min(\{v(x_i x_{i+1}) | 0 \leq i < k\}) | x_0 \dots x_k \in \mathcal{C}_{xy}\})$$

Question 1.1.2 En déduire, à l'aide d'arguments simples, que la chaîne de débit maximum entre I_6 et I_2 pour le graphe exemple a un débit égal à trois.

1.2

Soit T un arbre couvrant de G de poids maximum (le poids de T étant la somme, sur toutes les arêtes de T, des poids de ces arêtes). On appelle T_1 et T_2 les deux composantes que l'on obtient, à partir de T, en enlevant l'arête de poids minimum sur la chaîne de T entre x et y. Prouver que la valuation minimale de la chaîne de T joignant x et y vaut $D(x, y)$ pour le réseau G.

1.3

Donner un algorithme déterminant, dans un arbre quelconque, l'unique chaîne entre deux sommets donnés.

1.4

Quelle méthode peut-on appliquer pour déterminer, dans un graphe quelconque G, une chaîne de débit maximum entre deux sommets quelconques de G ?

1.5

Appliquer cette méthode pour déterminer une chaîne de débit maximum entre I_1 et I_4 dans le réseau exemple.

Exercice 2 – Réduction

On considère le problème suivant :

— **nom :** Couverture
— **entrée :** un graphe $G = (V, E)$ et un entier k
— **question :** existe-t-il $V' \subseteq V$ avec $|V'| \leq k$ tel que toute arête de G possède au moins une extrémité dans V' ?

2.1

Donner une couverture de cardinal minimum (justifier le) pour le graphe de la figure 2.

2.2

Montrez que le problème couverture est dans NP.

2.3

Montrez que le problème couverture est NP-complet (toutes les reductions du cours sont utilisables).

Exercice 3 – Degrés des somments d'un graphe

Le but de cet exercice est de donner des pistes pour déterminer si une suite de n entiers peut être vue comme les degrés d'un graphe à n sommets. Par exemple la suite $(1, 2, 2, 3, 3, 3, 4)$ admet le graphe de la figure 2 :

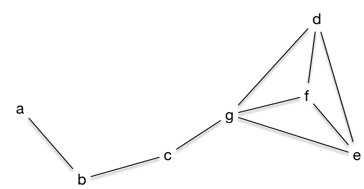

Figure 2 – Les sommets (a, b, c, d, e, f, g) ont respectivement pour degré $(1, 2, 2, 3, 3, 3, 4)$

3.1

Montrer que la somme des entiers est nécessairement paire pour qu'un tel graphe existe, mais que cette condition n'est pas suffisante.

3.2

Nous allons ici résoudre ce problème pour les graphes orientés grâce à la théorie des flots. Soit n un entier positif. On considère $2n$ entiers positifs ou nuls, notés $d_1^+, \dots d_i^+, \dots d_n^+$ et $d_1^-, \dots d_i^-, \dots d_n^-$. On cherche à construire un graphe **orienté** $G = (V, E)$ sans boucle ni d'arc multiple dont les sommets, notés $x_1, \dots, x_i \dots, x_n$, vérifient $\delta^+(x_i) = d_i^+$ et $\delta^-(x_i) = d_i^-$ pour $1 \leq i \leq n$,

Question 3.2.1 Modéliser le problème sous forme d'un problème de flot maximum. Exprimer à l'aide de n la complexité de l'algorithme de Ford et Fulkerson quand on l'applique à ce problème.

Résoudre le problème pour $n = 5$ et
— $d_1^+ = 2$, $d_2^+ = 3$, $d_3^+ = 1$, $d_4^+ = 1$, $d_5^+ = 3$,
— $d_1^- = 2$, $d_2^- = 1$, $d_3^- = 2$, $d_4^- = 2$, $d_5^- = 3$,

Question 3.2.2 Est-il possible d'adapter le modèle pour des graphes non dirigé ? Si oui, montrez la construction pour le graphe de la figure 2.

3.3

Vous aller caractériser les graphes à $n > k$ sommets tels que $\delta(x) = k$ pour tout sommet x. Ces graphes sont appelés k-régulier.

Question 3.3.1 Montrer qu'il est nécessaire que $k \cdot n$ soit pair pour qu'un graphe k-régulier existe.

Question 3.3.2 Montrez que si $n = 2m$ et $m \geq k$ on peut construire un graphe biparti k-régulier.

Question 3.3.3 Montrez que si $k = 2m$ on peut construire un graphe k-régulier pour tout $n \geq k + 1$.

Question 3.3.4 Montrer qu'il est suffisant que $k \cdot n$ soit pair pour qu'un graphe k-régulier existe.

Exercice 4 – Chemins dans un cylindre

On considère le problème suivant : On considère $n \cdot p$ entiers positifs a_{ij} $(1 \leq i \leq n, 1 \leq j \leq p)$, écrits sur un cylindre ayant n lignes et p colonnes, comme illustré figure 3.

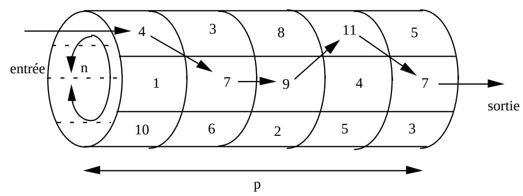

Figure 3 – Le cylindre

Un chemin est tracé de l'entrée du cylindre jusqu'à la sortie, avec la restriction que, d'une case, on ne peut aller qu'aux trois positions de la colonne suivante adjacentes à la position courante. Le coût d'un tel chemin est la somme des entiers écrits dans les cases traversées (par exemple, le chemin tracé sur le dessin a un coût égal à 38).

4.1

Combien de chemins distincts a-t-on de l'entrée à la sortie, sans imposer les cases de départ et d'arrivée ?

4.2

Expliciter un algorithme, prenant en entrée tous les a_{ij} $(1 \leq i \leq n, 1 \leq j \leq p)$, qui détermine un tel chemin de coût minimum en $\mathcal{O}(np)$ opérations. On justifiera le fait que l'algorithme calcule bien ce qu'il faut, ainsi que sa complexité.

4.3

L'appliquer à l'exemple de la figure 4 où le cylindre a été déplié ($n = 4, p = 5$; les bords ab et cd sont confondus), et en déduire un chemin de coût minimum.

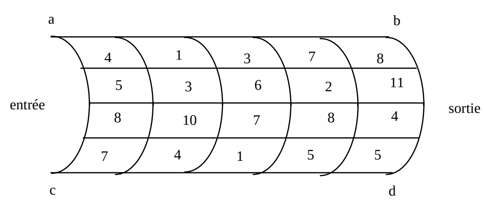

Figure 4 – Le cylindre (après aplatissement)

4.4

A l'aide de l'algorithme de Dijkstra, retrouver le chemin précédent. Du point de vue de la complexité, lequel des deux algorithmes est-il le plus avantageux d'appliquer pour résoudre ce problème ?