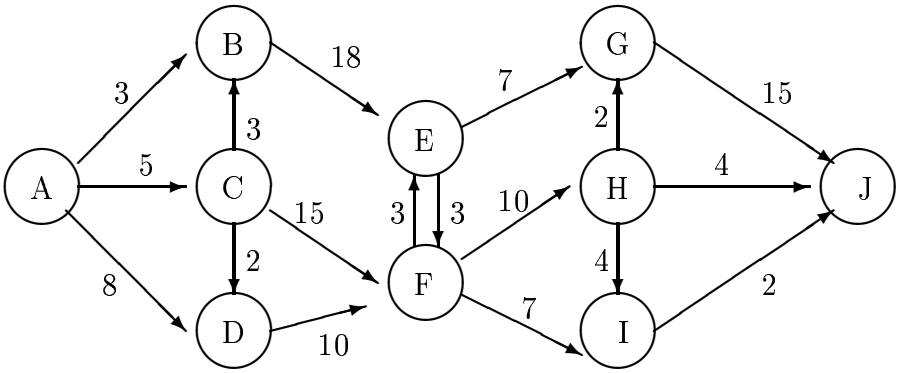
Algorithmique des Graphes – TD 10

Exercice 1

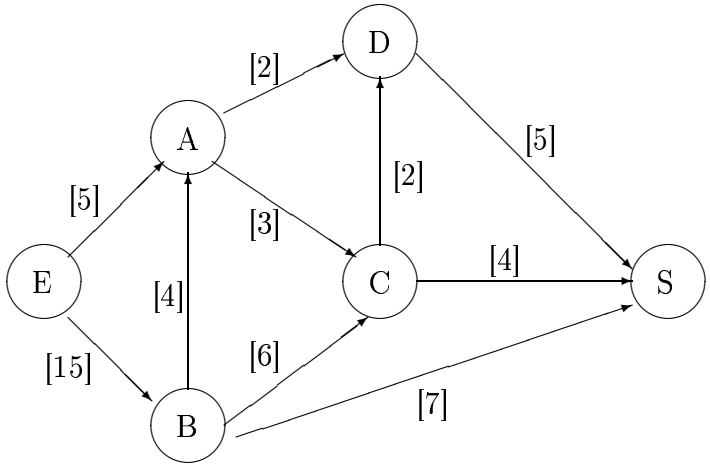
On considère le graphe orienté valué suivant :



1. En utilisant l'algorithme de Kruskal, déterminer un arbre recouvrant de poids minimal du graphe non orienté sous-jacent. Combien y a-t-il de solutions possibles ? (Justifier la réponse.)
2. Indiquer un chemin de poids minimal allant du sommet A au sommet J (on précisera l'algorithme utilisé, et on en indiquera les différentes étapes).
3. Indiquer un chemin de poids minimal allant du sommet A au sommet J en passant par le sommet E (on expliquera comment ce chemin a été construit).
4. On suppose à présent que les valuations des arcs représentent des capacités. Déterminer un flot maximal allant du sommet A au sommet J en indiquant les différentes étapes de l'algorithme utilisé ; indiquer aussi une coupe minimale.

Exercice 2

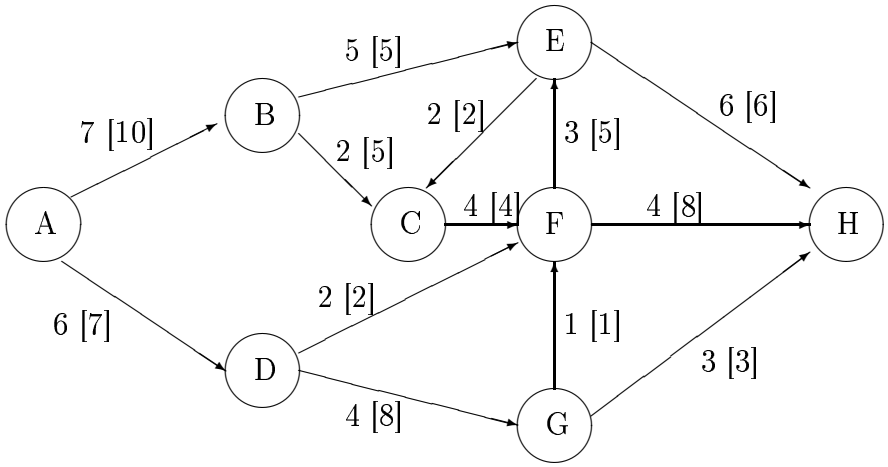
Les valuations des arcs du graphe suivant représentent des capacités :



En utilisant l'algorithme de Ford-Fulkerson, déterminer sur ce graphe un flot maximal du sommet E au sommet S (on indiquera les augmentations successives du flot, et on mettra en évidence le fait que le dernier flot obtenu est maximal).

Exercice 3

Dans la figure suivante, les valuations des arcs définissent un flot sur un graphe borné (les capacités des arcs sont indiquées entre crochets) :



Le flot de A à H est-il maximal ? Sinon, l'optimiser.

On indiquera une coupe minimale associée au flot maximal obtenu.

Exercice 4

On s'intéresse à la *réduction* de divers problèmes au problème du flot maximum dans un réseau.

PB1 (remplissage de tableau) :

Donnée :

$m, n \in \mathbb{N}; c'_1, c'_2, \ldots, c'_m, c''_1, c''_2, \ldots, c''_n \in \mathbb{N}$;

Calculer :

Des nombres $x_{i,j} \in \mathbb{N}$, pour $1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n$, tels que :

(R1) pour toute ligne d'indice $i \in [1, m], \sum_{j=1}^{n} x_{i,j} \leqslant c'_i$

(R2) pour toute colonne d'indice $j \in [1, n], \sum_{i=1}^{m} x_{i,j} \leqslant c''_j$

(R3) la somme totale $\sum_{i=1}^{m} \sum_{j=1}^{n} x_{i,j}$ soit maximale (parmi les sommes des familles $x_{i,j}$ vérifiant (R1) et (R2)).

Considérons le réseau suivant :

$R = (S, A, c)$ où

$$S = \{p, s\} \cup \{L_1, L_2, \ldots, L_m\} \cup \{C_1, C_2, \ldots, C_n\}$$

$$A = \{(s, L_i) \mid 1 \leqslant i \leqslant m\} \cup \{(C_j, p) \mid 1 \leqslant j \leqslant n\} \cup \{(L_i, C_j) \mid 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n\}$$

et la fonction de capacité c, est donnée par :

$$c(p, s) = c(L_i, C_j) = +\infty \quad (\text{ pour tous } i, j)$$

$$c(s, L_i) = c'_i \quad (\text{ pour } 1 \leqslant i \leqslant m)$$

$$c(C_j, p) = c''_j \quad (\text{ pour } 1 \leqslant j \leqslant n).$$

2-Montrer que l'on peut associer à tout flot admissible sur R, un remplissage $x_{i,j}$ satisfaisant les conditions (R1) et (R2), tel que

$$c(p, s) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{i,j}$$

3- En déduire un algorithme résolvant PB1 , qui soit fondé sur l'algorithme de Ford et Fulkerson (de calcul d'un flot maximal sur un réseau).

On considère maintenant le

PB2 (problème des cases admissibles) :

Donnée :

$m, n \in \mathbb{N}; c'_1, c'_2, \ldots, c'_m, c''_1, c''_2, \ldots, c''_n \in \mathbb{N}$;

$A_{i,j} \in \{0, 1\}$, pour $1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n$

Calculer :

Des nombres $x_{i,j} \in \mathbb{N}$, pour $1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n$, tels que :

(R0) si $A_{i,j} = 0$ alors $x_{i,j} = 0$

(R1) pour toute ligne d'indice $i \in [1, m], \sum_{j=1}^{n} x_{i,j} \leqslant c'_i$

(R2) pour toute colonne d'indice $j \in [1, n], \sum_{i=1}^{m} x_{i,j} \leqslant c''_j$

(R3) la somme totale $\sum_{i=1}^{m} \sum_{j=1}^{n} x_{i,j}$ soit maximale (parmi les sommes de toutes les familles $x_{i,j}$ vérifiant (R0)(R1) et (R2)).

En d'autres termes : il s'agit encore de remplir un tableau $m \times n$, mais cette fois-ci, toutes les cases (i, j) telles que $A_{i,j} = 0$ doivent avoir un contenu nul.

4- Construire, pour toute donnée du PB2, un réseau $R = (S, A, c)$, de façon à réduire PB2 au problème du flot maximum.

5- Résoudre, par cette méthode, le PB2 sur la donnée :

$m = 3, n = 4, (c'_1, c'_2, c'_3) = (6, 4, 2), (c''_1, c''_2, c''_3, c''_4) = (1, 6, 3, 2)$ et

$A_{i,j} = 0 \Leftrightarrow (i, j) \in \{(2, 1), (2, 2), (3, 1), (3, 2), (3, 3)\}$

On considère maintenant le

PB3 (problème du couplage maximum) :

Donnée :

Un graphe $G = (S, A)$, orienté, qui est *biparti*, i.e. tel que S admet une partition en $S = S_1 \cup S_2$ $(S_1 \cap S_2 = \emptyset)$, telle que $A \subseteq S_1 \times S_2$.

Calculer :

Un ensemble d'arcs $A' \subseteq A$ tel que :

(C1)tout sommet de S_1 (resp. S_2) est le début (resp. l'extrémité) d'au plus un arc de A'

(C2)le nombre d'éléments de A' est maximal parmi tous les ensembles A' satisfaisant (C1).

(Une partie A' vérifiant (C1) est un *couplage*).

6- Construire, pour toute donnée du PB3, un réseau $R = (S, A, c)$, de façon à réduire PB3 au problème du flot maximum.

7- Résoudre, par cette méthode, le PB3 sur la donnée :

$S_1 = \{1, 2, 3\}, S_2 = \{4, 5, 6\}, A = \{(1, 5), (2, 4), (2, 5), (2, 6), (3, 5)\}$.

8-On admet que l'une des versions de l'algorithme de Ford et Fulkerson a une complexité polynomiale par rapport à la taille du réseau R (en admettant que les opérations arithmétiques sur les entiers ont une complexité constante). Montrer que les problèmes PB1,PB2,PB3 peuvent être résolus en temps polynomial.