Domaine Sciences et Technologies
LICENCE 3 INFORMATIQUE

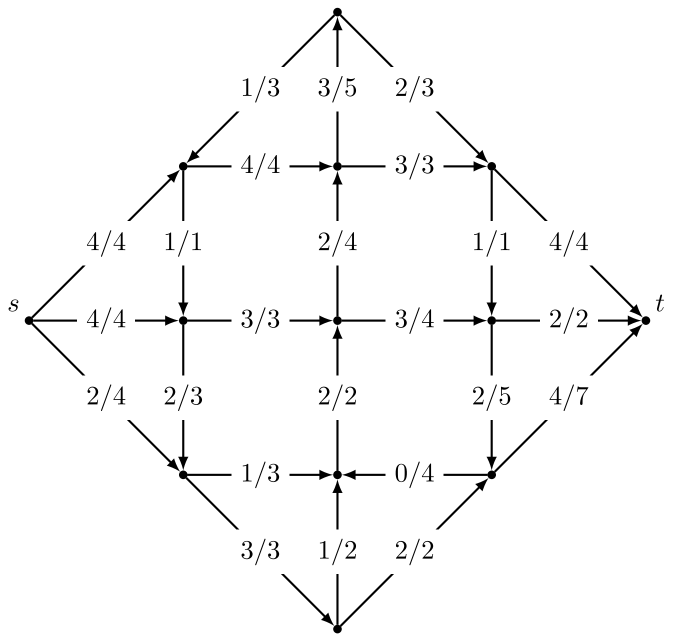
**Algorithmique 2 : TD 2**
**Code UE : SIN5U03C**

Année 2018-19

**Flot maximum**

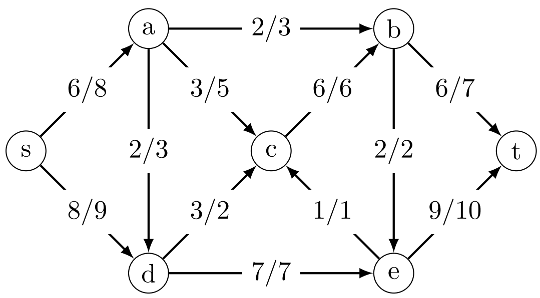
**Exercice 1** (Graphe résiduel)

Construire le graphe résiduel pour le $(s,t)$-flot suivant (flot/capacité). Le flot est-il maximum ? Trouver une $(s,t)$-coupe de capacité minimum.

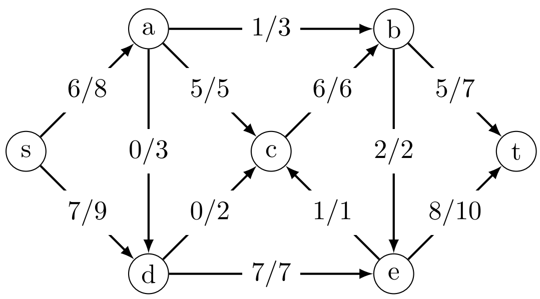



**Exercice 2** (Encore des flots)

(a) Lister les raisons pour lesquelles les valeurs données dans le graphe suivant ne représentent pas un $(s,t)$-flot (la notation $f/c$ permet de représenter la valeur $f$ du flot et la capacité $c$ d'un arc).




(b) Calculer un $(s,t)$-flot maximum pour ce graphe, et une $(s,t)$-coupe minimum. Vous pouvez vous aider du flot suivant.



(c) Le flot maximum augmenterait-il si la capacité de l'arc $(d, c)$ augmentait ? Justifier par un argument le plus simple possible.

**Exercice 3** (Flot maximum avec plusieurs sources et puits) Réduire le problème de flot maximum avec plusieurs sources $s_1, \ldots, s_k$ et plusieurs puits $t_1, \ldots, t_m$ au problème de *flot maximum classique* (avec une seule source $s$ et un seul puits $t$).

**Exercice 4** (Nombre des coupes min) Décrire un graphe à $n$ sommets avec (a) une seule coupe minimum, (b) $O(n)$ coupes minimum, (c) $O(n^2)$ coupes minimum, (d) $O(n^3)$ coupe minimum.

**Exercice 5** (Couplage dans un graphe biparti) Un graphe est dit *biparti* si son ensemble de sommets peut être divisé en deux sous-ensembles disjoints $A$ et $B$ tels que chaque arête ait une extrémité dans $A$ et l'autre dans $B$. Un *couplage* d'un graphe est un ensemble d'arêtes de ce graphe qui n'ont pas de sommets en commun. Un *couplage maximum* est un couplage contenant le plus grand nombre possible d'arêtes.

Modeliser le problème de couplage maximum dans un graphe biparti $G = (A \uplus B, E)$ comme un problème de flot maximum dans un réseau. Deffinisez l'ensemble de sommets, l'ensemble d'arcs et leur capacités.

**Exercice 6** ($(s, t)$-Connexité) Étant donnés un graphe dirigé $G = (V, A)$ et deux sommets $s$ et $t$ de $G$, comment calculer la connexité entre $s$ et $t$ (i.e. le nombre maximum des chemins arête-disjoints entre $s$ et $t$) ? Réduire ce problème à un *problème de flot maximum.* Même question lorsque $G$ est un graphe non-dirigé.

**Exercice 7** Soit $N = (V, A, s, t, c)$ un réseau de transport avec $n$ sommets, $m$ arcs, et muni d'une fonction de capacités $c$ prenant des valeurs entières. On dit que $N$ est un réseau *à flot maximum unique* (respectivement, *à coupe minimum unique*) si $N$ admet un seul flot maximum (respectivement, un seule coupe coupe minimum).

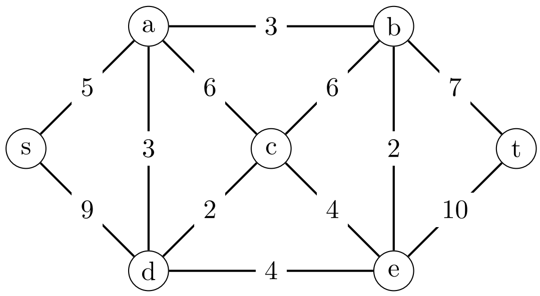
1. Donnez un exemple de réseau de transport à flot maximum unique qui n'est pas un réseau à coupe minimum unique.
2. Donnez un exemple de réseau de transport à coupe minimum unique qui n'est pas un réseau à flot maximum unique.
3. Étant donné un réseau de transport $N$, décrire et justifier un algorithme pour déterminer si $N$ est un réseau à flot maximum unique.
4. Étant donné un réseau de transport $N$, décrire et justifier un algorithme pour déterminer si $N$ est un réseau à coupe minimum unique.

*Indication :* Pour les deux dernières questions, formuler au plus $m$ problèmes de flot maximum en modifiant la capacité de certains arcs du réseau N.

**Exercice 8** (Trouver un plus large chemin)

Soit $N = (V, A, s, t, c)$ un réseau de transport non-orienté avec $n$ sommets, $m$ arêtes, ou $c(u, v)$ peut être interprété comme la bande passante de l'arête $uv$. La *largeur* $\alpha(P)$ d'un $(s, t)$-chemin $P = (s = v_0, v_1, v_2, \ldots, v_k, v_{k+1} = t)$ est le minimum des capacités de ses arêtes : $\alpha(P) = \min\{c(v_i, v_{i+1}) : i = 0, \ldots, k\}$. Un *plus large* $(s, t)$*-chemin* est un $(s, t)$-chemin de largeur maximum. Notons par $\alpha^*$ la largeur d'un plus large $(s, t)$-chemin : $\alpha^* = \max\{\alpha(P) : P \text{ chemin entre } s \text{ et } t\}$.

(1) Calculer $\alpha^*$ ainsi qu'un plus large chemin $P^*$ pour le réseau suivant :




Pour une $(s,t)$-coupe $(S,\overline{S})$ soit $A(S,\overline{S})$ son ensemble d'âretes. La *finesse* $\beta(S,\overline{S})$ d'une $(s,t)$-coupe $(S,\overline{S})$ est le maximum des capacités de ses arêtes : $\beta(S,\overline{S}) = \max\{c(uv) : uv \in A(S,\overline{S})\}$. Une *plus fine* $(s,t)$*-coupe* est une $(s,t)$-coupe de finesse minimale. Notons par $\beta^*$ la finesse d'une plus fine $(s,t)$-coupe : $\beta^* = \min\{\beta(S,\overline{S}) : (S,\overline{S}) \text{ coupe entre } s \text{ et } t\}$.

(b) Calculer $\beta^*$ ainsi qu'une coupe la plus fine $(S^*,\overline{S^*})$ pour le réseau d'avant.

(c) Montrez que pour tout $(s,t)$-chemin $P$ et pour toute $(s,t)$-coupe $(S,\overline{S})$, l'inégalité suivante est vrai : $\alpha(P) \leq \beta(S,\overline{S})$. Déduire l'inégalité $\alpha^* \leq \beta^*$.

(d) Pour démontrer l'inégalité inverse, procédez de façon algorithmique, comme dans l'algorithme de Ford-Fulkerson. Notamment, décrire un algorithme pour calculer un $(s,t)$-chemin $P^*$ et une $(s,t)$-coupe $(S^*,\overline{S^*})$ tels que $\alpha(P^*) = \beta(S^*,\overline{S^*})$. Conclure que $\alpha^* = \beta^*$.

(e) Quelle est la complexité de cet algorithme ?