Exercice 1. Terminaison de Ford-Fulkerson

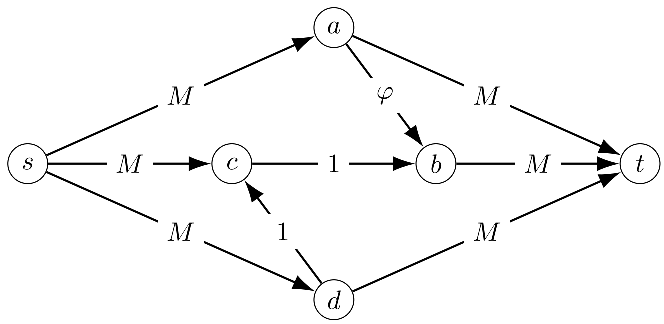
1. Rappeler pourquoi l'algorithme de Ford-Fulkerson termine lorsque les capacités sont entières.

   > **Solution** : voir cours

2. Montrer que l'algorithme de Ford-Fulkerson termine lorsque les capacités sont rationnelles.

   > **Solution** : Soit q le produit de tous les dénominateurs des capacités. Multiplions toutes les capacités par q, de façon à obtenir des capacités entières. Ford-Fulkerson procède de la même façon sur ce nouveau graphe (comme tout est multiplié par q, ce seront les mêmes chemins qui sont trouvés à chaque itération et la même proportion de capacité résiduelle retirée). Comme l'algorithme termine pour des capacités entières, il termine aussi pour des capacités rationnelles.

3. Montrer que l'algorithme de Ford-Fulkerson peut ne pas terminer avec le graphe ci-dessous, où $\varphi = \dfrac{1+\sqrt{5}}{2}$ est la solution positive de l'équation $x^2 - x - 1 = 0$ et M un grand entier.



Exercice 2. Application à la connectivité
On dit que des chemins sont **arc-disjoints** s'ils n'ont pas d'arc en commun. Soit $\overrightarrow{G} = (V, \overrightarrow{E})$ un graphe orienté (sans capacité) et u, v deux sommets de $\overrightarrow{G}$. La **connectivité** de u à v est le nombre maximum de chemins arc-disjoints de u à v, dans $\overrightarrow{G}$.

1. Expliquer comment utiliser l'algorithme de Ford-Fulkerson pour déterminer la connectivité de u à v.

   > **Solution** : On met une capacité 1 sur chaque arête et on cherche un flot maximum de $s = u$ à $t = v$. La valeur maximum d'un flot est alors la connectivité de u à v.

2. Comment faire pour obtenir en plus les chemins arc-disjoints correspondants ?

   > **Solution** : On peut obtenir un flot optimum f^* puis appliquer le théorème de décomposition du cours pour obtenir des chemins C_1, ..., C_n arc-disjoints de u à v. S'il était possible d

3. Prouver le **théorème de Menger** : la connectivité de u à v est égal au nombre minimum d'arc qu'il faut enlever de $\overrightarrow{G}$ pour déconnecter u et v (c'est à dire : pour qu'il n'existe plus de chemin de u à v).

La **connectivité** de $\overrightarrow{G}$ est la connectivité minimum d'un sommet quelconque de $\overrightarrow{G}$ à un autre.

4. Montrer que $\overrightarrow{G}$ a un sommet s de degré sortant inférieur ou égal à $\dfrac{|\overrightarrow{E}|}{|V|}$.

   > **Solution** : $\dfrac{|\overrightarrow{E}|}{|V|}$ est le degré sortant moyen des sommets de $\overrightarrow{G}$: il existe un sommet dont le degré sortant est supérieur ou égal à la moyenne.

5. En déduire que la connectivité de $\overrightarrow{G}$ peut être déterminée en $\mathrm{O}(\overrightarrow{E}^2)$.

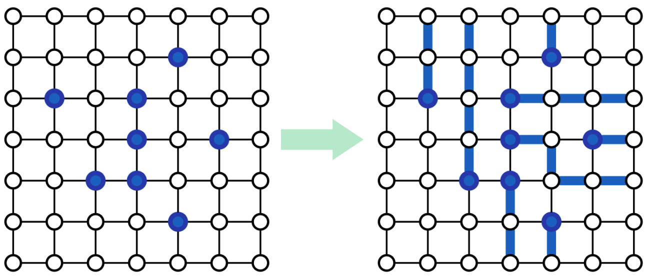
   > **Solution** : Soit $n = |V|$, $p = |\overrightarrow{E}^2|$ et u et v des sommets tels que la connectivité c de $\overrightarrow{G}$ soit celle de u à v. D'après III.3, c'est aussi la capacité minimum d'une coupe S séparant u et v. Si $s \in S$ alors une utilisation de utilisation de l'algorithme de Ford-Fulkerson avec $0 = s$ et $1 = v$ donnera c. Si $s \notin S$ alors une utilisation de utilisation de l'algorithme de Ford-Fulkerson avec $0 = u$ et $1 = s$ donnera c. Il suffit donc d'appliquer Ford-Fulkerson avec $0 = s$ et pour tout autre sommet 1, puis appliquer Ford-Fulkerson avec $1 = s$ et pour tout autre sommet 0.
   > Le maximum des valeurs trouvées est alors c. De plus chaque utilisation de Ford-Fulkerson effectue $\mathrm{O}(\frac{p}{n})$ recherche de chemin car chaque arc sortant de s ne peut être utilisé qu'une seule fois. Comme chaque recherche de chemin est en $\mathrm{O}(p)$ (avec parcours en largeur ou profondeur), chaque utilisation de Ford-Fulkerson est en $\mathrm{O}(\frac{p^2}{n})$.
   > Cette méthode effectue n appels à Ford-Fulkerson, d'où une complexité totale $\boxed{\mathrm{O}(p^2)}$.

6. Comment adapter cet algorithme si on remplace arc-disjoints par sommets-disjoints (2 chemins ne doivent pas avoir de sommets en communs) ?

   > **Solution** : On remplace chaque sommet v par deux sommets v_1 et v_2. On remplace chaque arc entrant dans v par un arc entrant dans v_1, et chaque arc sortant de v par un arc sortant de v_2. On met une capacité infinie (ou une capacité très grande, par exemple égale au nombre d'arcs total) sur ces arcs. On met de plus un arc de de v_1 et v_2 de capacité 1.
   > On applique enfin Ford-Fulkerson pour obtenir des chemins arc-disjoints. On remplace alors $u \longrightarrow v_1 \longrightarrow v_2 \longrightarrow w$ par $u \longrightarrow w$ dans le graphe initial.
   > Ceci nous donne des chemins sommet-disjoints dont le nombre est maximum.

7. On considère maintenant un jeu d'escape game sur une grille $n \times n$, où p sommets sont sélectionnés. Décrire une

méthode pour déterminer s'il existe des chemins sommet-disjoints de chaque sommet sélectionné vers le bord de la grille.



Un exemple de configuration initiale (gauche) et une résolution (droite)

> **Solution** : On ajoute deux sommets s et t. On relie s à chaque sommet selectionné, et on relie chaque sommet du bord vers t. Puis on utilise la question précédente.

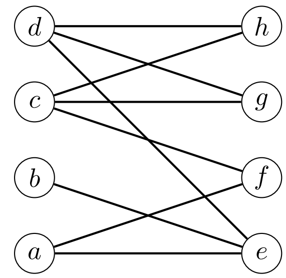
Exercice 3. Application au couplage maximum

Un **couplage** dans un graphe non-orienté est un ensemble M d'arêtes dont toutes les extrémités sont différentes (si $e_1 \in M$ et $e_2 \in M$ alors $e_1 \cap e_2 = \emptyset$). Un couplage avec un nombre maximum d'arêtes est un **couplage maximum**.

1. Soit $G = (V, E)$ un graphe non-orienté **biparti**, c'est à dire que $V = A \sqcup B$ et toute arête de G a une extrémité dans A et une dans B. Expliquer comment utiliser l'algorithme de Ford-Fulkerson pour trouver un couplage maximum dans G.

   > **Solution** : on oriente les arêtes de G de A vers B, on ajoute un sommet s, un sommet t à G et on donne une capacité 1 à chaque arc.
   > On cherche ensuite un flot maximum f^* dans ce graphe avec Ford-Fulkerson. Comme les capacités initiales sont des entiers et que Ford-Fulkerson ajoute à chaque étape un flot égale à l'une des capacité, f^* est un flot entier (c'est à dire : $f^*(\overrightarrow{e})$ est un entier pour tout arc $\overrightarrow{e}$).
   > On conserve ensuite l'ensemble M des arêtes entre A et B dont le flot est égal à 1. Soit $v \in A$. Alors, comme le flot entrant dans v est au plus la capacité de (s, v) qui est égale à 1, il peut y avoir au plus un arc sortant de v avec un flot de 1. De même pour les sommets de B.
   > Ainsi, deux arêtes de M ne peuvent pas contenir le même sommet : M est bien un couplage.
   > Il est de taille maximum car sinon, on pourrait inversement en déduire un flot de valeur plus grande que f^*, ce qui est absurde.

2. En utilisant cette méthode, trouver un couplage maximum dans le graphe suivant :



Exercice 4. Flot maximum de poids minimum

On considère maintenant un poids $w : \overrightarrow{E} \longrightarrow \mathbb{R}^+$ sur chaque arc. Le poids (ou coût) $w(f)$ d'un flot f est défini par $w(f) = \sum_{\overrightarrow{e} \in \overrightarrow{E}} w(\overrightarrow{e}) f(\overrightarrow{e})$. Un **flot maximum de poids minimum** est un flot de valeur maximum dont le poids est minimum parmi tous les flots de valeur maximum.

1. Soit f un flot au cours de l'algo. de Ford-Fulkerson. Supposons que le graphe résiduel contienne un cycle de poids négatif. Comment modifier f de façon à obtenir un flot de même valeur mais de poids strictement inférieur?

2. Comment détecter un cycle de poids négatif dans un graphe pondéré?

3. Proposer un algorithme pour trouver un flot maximum de poids minimum.