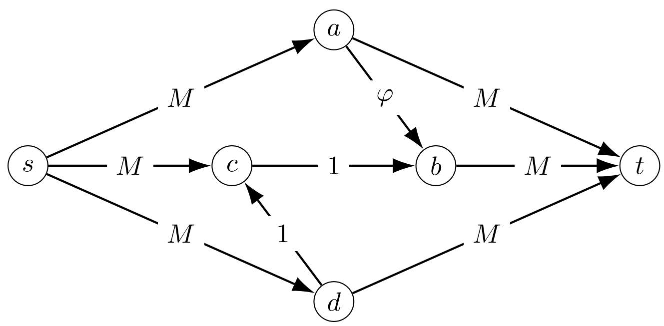
Exercice 1. Terminaison de Ford-Fulkerson

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



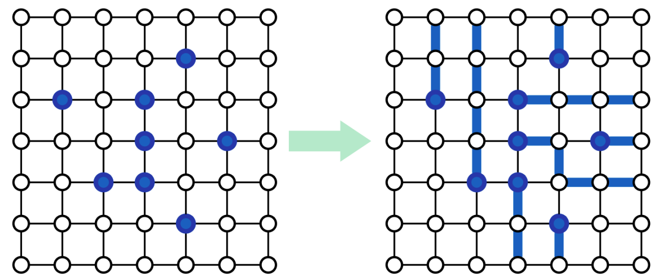
Exercice 2. Application à la connectivité

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.
2. Comment faire pour obtenir en plus les chemins arc-disjoints correspondants ?
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 $\dfrac{|\overrightarrow{E}|}{|V|}$.
5. En déduire que la connectivité de $\overrightarrow{G}$ peut être déterminée en $O(\overrightarrow{E}^2)$.
6. Comment adapter cet algorithme si on remplace arc-disjoints par sommets-disjoints (2 chemins ne doivent pas avoir de sommets en communs) ?
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.

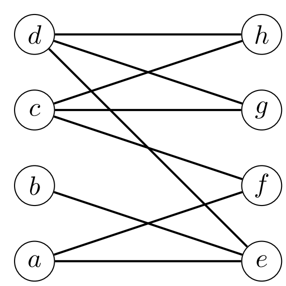


Un exemple de configuration initiale (gauche)
et une résolution (droite)

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.
   <u>Indice</u>: on pourra orienter les arêtes de G, rajouter un sommet s et un sommet t à G et donner une capacité à chaque arc.
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 $w(f) = \displaystyle\sum_{\vec{e} \in \overrightarrow{E}} w(\vec{e}) f(\vec{e})$. 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.