

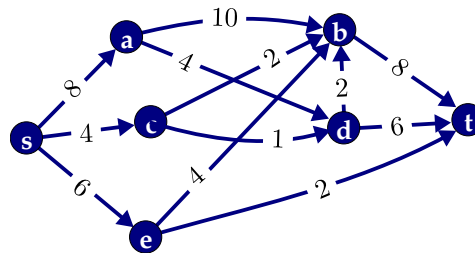
# RO202 - Initiation à la Recherche Opérationnelle

Zacharie Ales, Tiphaine George  
2025 - 2026

## EXERCICES 2 - Flots

### Exercice 1 Ford-Fulkerson

On donne le graphe avec capacités suivant :



1. Trouver un flot complet (en prenant obligatoirement les sommets selon l'ordre alphabétique).
2. Donner une borne supérieure de la valeur d'une coupe minimale (considérer les arcs incidents à s puis à t).
3. Appliquer l'algorithme de Ford-Fulkerson pour trouver un flot maximal (en marquant obligatoirement les sommets selon l'ordre alphabétique) et en déduire la coupe minimale.
4. On a la possibilité d'augmenter la capacité d'un arc. Lequel choisir pour pouvoir augmenter le flot au maximum ?

### Exercice 2 Implémentation de Ford-Fulkerson

1. Inclure à votre projet java RO202 le fichier `fordFulkerson.py`.
2. Compléter la méthode `def fordFulkerson()` permettant d'appliquer l'algorithme de Ford-Fulkerson à un graphe `g` entre une source `s` et un puits `t`.  
*Remarque :* Pour simplifier, vous commencerez avec un flot initial nul plutôt que complet.
3. Vérifier que votre méthode retourne le même résultat que celui obtenu à l'exercice précédent.
4. Utiliser cette méthode pour résoudre le problème de flot maximal des graphes représentés en Figure 1.

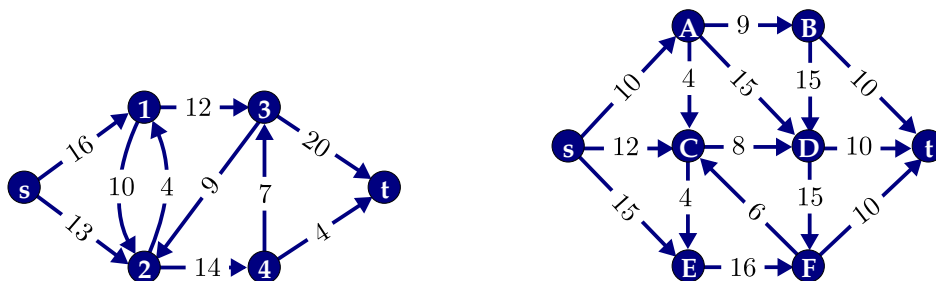


FIGURE 1 – Graphes pour lesquels on cherche un flot maximal.

### Exercice 3 Couplage

Un coffre-fort est composé de  $r$  serrures qu'il faut déverrouiller simultanément pour l'ouvrir. On dispose de  $l$  clefs de sorte que toute clef ouvre au moins une serrure, une serrure peut être ouverte par au moins une clef et  $l \geq r$ . Sachant quelles serrures chaque clef ouvre, on se demande si l'on

peut ouvrir le coffre. On représente la situation à l'aide d'un graphe biparti  $G = (V_1, V_2, A)$  où les sommets de  $V_1$  sont les clefs, les sommets de  $V_2$  sont les serrures et il y a un arc dans  $A$  d'un sommet clef de  $V_1$  vers et un sommet serrure de  $V_2$  si l'une permet d'ouvrir l'autre. Proposez une méthode générale qui permet d'obtenir une solution s'il en existe une, et montre comment déverrouiller un maximum de serrures dans le cas contraire.

#### Exercice 4 Chemins disjoints

1. Donner une méthode générale permettant de trouver un nombre maximum de chemins arcs-disjoints entre un sommet source  $s$  et une destination  $t$  dans un réseau.
2. Appliquez la méthode pour le réseau ci-dessous (d'abord à la main, puis en utilisant votre implémentation de l'algorithme de Ford-Fulkerson).
3. Comment faire si on veut en plus que les chemins soient sommets-disjoints ?
4. Appliquez cette seconde méthode pour le réseau ci-dessous (d'abord à la main, puis en utilisant votre implémentation de l'algorithme de Ford-Fulkerson).

