# INF2010 – Structures de données et algorithmes
# Hiver 2018
# Travail Pratique 6
# Graphes

**Objectifs :**

- Détecter s'il existe un cycle dans un graphe orienté.
- Maîtriser le parcours de graphe en profondeur (DFS)
- Trouver l'arbre sous-tendant minimum (MST) d'un graphe.

## Exercice 1 : Détection de cycle à l'aide d'un parcours en profondeur (1.5)

Pour donner suite à un discours très motivant de votre professeur, vous décidez de vous intéresser à son cours de compilateur et entamer l'écriture d'un interpréteur. En tant que développeur aux bonnes pratiques de programmation, vous désirez pouvoir réutiliser votre code et mettez ainsi en place un système d'importation de fichiers de code source. Cependant, vous réalisez rapidement qu'il se peut qu'un projet contienne des dépendances circulaires et que cela mènerait à une boucle infinie lors de l'interprétation. Afin d'y remédier, vous décidez d'appliquer vos connaissances sur les graphes de manière à détecter les dépendances circulaires.

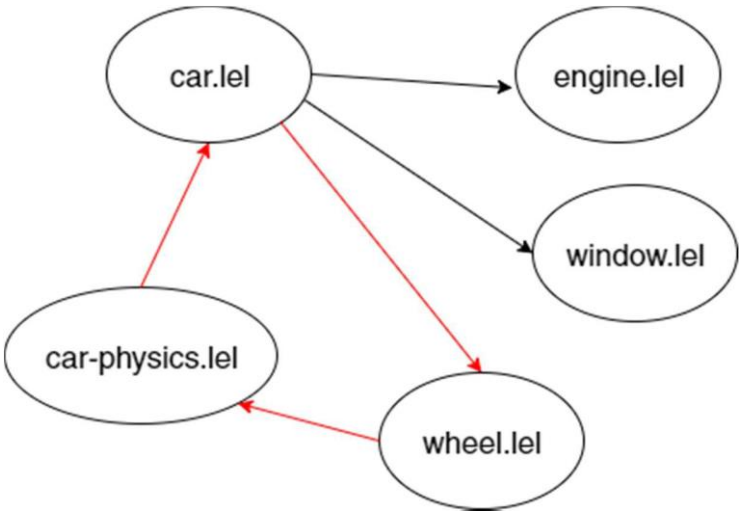


Figure 1 : Graphe formé par les fichiers de code source (les arrêtes en rouge montrent une dépendance circulaire

Le premier exercice consiste donc à utiliser un parcours en profondeur (DFS) sur un graphe dirigé afin de détecter s'il existe un cycle dans ce dernier. Pour ce faire, vous devez compléter le fichier CodeBase.java. Le fichier CircularDependenciesMain.java vous permettra ensuite de tester votre implémentation.

## Exercice 2 : Arbre sous-tendant minimum (3.5)

Le Vin et Fromage du CEGInfo-CEGL ayant été un franc succès, vous êtes parvenu à vous dénicher un stage chez le tout nouveau fournisseur d'accès Internet PolyNet. Lors de votre première semaine, on vous demande de mettre au point un logiciel permettant de calculer la quantité de câble minimale nécessaire afin de relier différentes villes où PolyNet souhaite offrir ses services. Vous mettez donc en pratique vos connaissances nouvellement acquises sur les graphes et décidez d'employer l'algorithme de Prim afin de trouver le minimum spanning tree (MST).

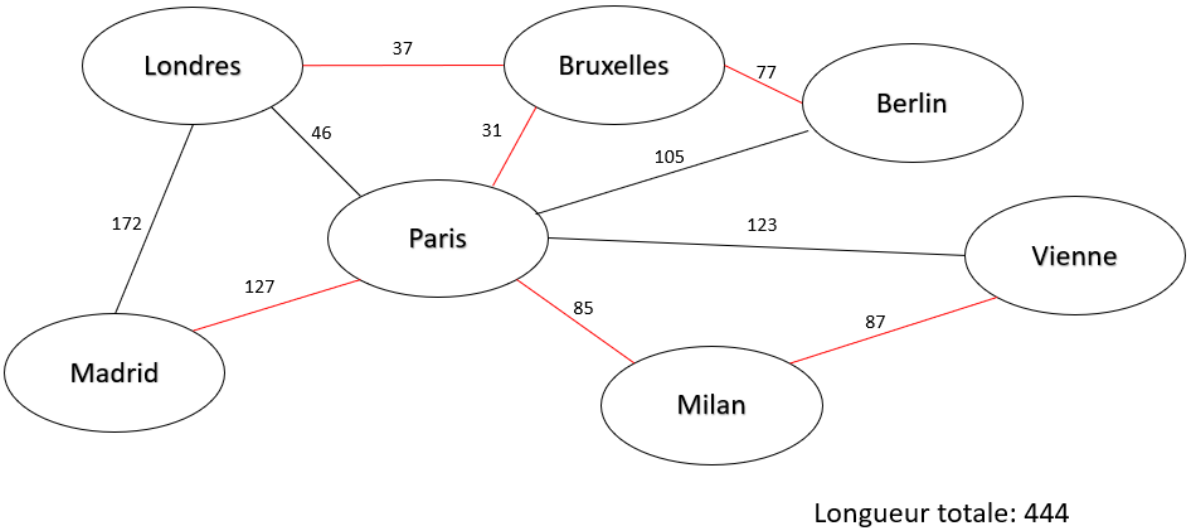


Figure 2 : Graphe représentant les différentes villes ainsi que la longueur du chemin entre chacune d'elles (les arrêtes en rouge forment l'arbre sous-tendant minimum).

Pour le second exercice, il vous est donc demandé d'implémenter l'algorithme de Prim afin de trouver l'arbre sous-tendant minimum dans un graphe. Pour ce faire, vous devez compléter le fichier PolyNet.java. Le fichier PolyNetMain.java vous permettra ensuite de tester votre travail.

## Instructions pour la remise

**** ATTENTION : Vous n'avez qu'une semaine pour compléter ce travail. ****

Le travail doit être fait par équipe de 2 personnes et doit être remis via Moodle au plus tard le :

Groupe 1 (jeudi B1) : Mercredi 18 avril 2018 à 23h55
Groupe 2 (jeudi B2) : Mercredi 11 avril 2018 à 23h55

Veuillez envoyer vos fichiers .java seulement, dans un seul répertoire contenu dans une archive de type *.zip uniquement, qui portera le nom : *inf2010_lab6_MatriculeX_MatriculeY.zip*, où MatriculeX < MatriculeY.

Les travaux en retard seront pénalisés de 20 % par jour de retard. Aucun travail ne sera accepté après 4 jours de retard. Si votre dépôt ne respecte pas la nomenclature définie ci-dessus, 0.5 point de pénalité sera appliqué.