

# MTH6412B : Projet voyageur de commerce (phase 1)

**Date:** Lundi 19 septembre 2016  
**auteur:** Dominique Orban  
**pondération:** 5%  
**remise:** le 26 septembre 2016

## Objectif

La première partie du projet consiste à définir de bonnes structures de données pour le problème du voyageur de commerce symétrique. Cette première partie est cruciale pour toute la suite.

## Marche à suivre

1. Récupérer le code de la phase 1 sur le site web du cours. Vous y trouverez une structure de données `Node` pour les noeuds d'un graphe et `Graph` pour le graphe proprement dit. On fournit également un jeu de fonctions qui lisent les fichiers au format de `TSPLib`. On se limite ici aux problèmes symétriques dont les poids sont donnés au format `EXPLICIT`.
2. Proposer une classe `Edge` pour représenter les arêtes d'un graphe.
3. Étendre la classe `Graph` afin qu'un graphe contienne ses arêtes. On se limite ici aux graphes non orientés. L'utilisateur doit pouvoir ajouter une arête à la fois à un graphe.
4. Étendre la méthode d'affichage (`__repr__`) d'un objet de type `Graph` afin que les arêtes du graphe soient également affichées.
5. Étendre la fonction `read_edges()` de `read_stsp.py` afin de lire les poids des arêtes (ils sont actuellement ignorés).
6. Fournir un programme principal qui lit une instance de TSP *symétrique* dont les poids sont donnés au format `EXPLICIT` et construit un objet de type `Graph` correspondant.

## Directives

- Écrire du code **lisible**, aéré et commenté. On pourra se reporter aux lignes directrices pour la rédaction de code Python à l'adresse <http://www.python.org/dev/peps/pep-0008/>.
- Utilisez les constructions de programmation orientée objet autant que possible. Vous en tirerez profit par la suite. Le degré d'utilisation des constructions de programmation orientée objet entre en ligne de compte dans la notation.