Rapport TP1 MTH6412B

Adrien Logut (1815142)

Choix d'implémentation

Edges

La première étape était de créer une classe Edge qui correspond aux arêtes du graphe. Dans notre implémentaion, on a fait le choix de faire des arêtes orientées (avec donc un début et une fin). Cependant dans le cas où l'on voudrait seulement des graphes non orientés, on peut rajouter un attribut booléen dans le modèle du graphe pour qu'il puisse tester les 2 sens (départ->fin et fin->départ).

Une arête est donc définie par :

— Un noeud de départ

— Un noeud d'arrivée

— Un poids

On a rajouté aussi des exceptions dans le cas où l'on voudrait créer des arêtes qui ne sont pas légales (Il manque une extrémité ou une arête qui boucle sur un noeud avec un poids nul).

Ces exceptions sont définies dans le fichier edgeException.py et permettent un code plus robuste. Ces exceptions sont testées si l'on execute le script tout seul.

Graph

Toutes les fonctions disponibles pour les noeuds(ajouter, obtenir la liste, obtenir le nombre) ont été implémentées pour les arêtes aussi. La fonction *repr* a été étendue pour afficher les arêtes du graphe aussi.

Main

Pour le programme principal, on a repris le code principal de read_stsp.py pour l'ouverture et la lecture des données. Ensuite les noeuds sont ajoutés. S'ils ont été définis de manière explicite, l'ajout se fait selon ces données. S'ils ne sont pas définis, le programme se charge de créer des noeuds d'id 0 à $dim - 1$, sans données accrochées.

Pour l'ajout d'arête, on vérifie juste que les arêtes n'ont pas de poids nul s'ils ont les mêmes extrémités.