Rapport SAE Exploration algorithmique S2.2

* FUCHS Thomas
* COMTE Gabriel

Partie 1 - Représentation d’un graphe

Dans cette première partie nous nous sommes familiarisés avec la représentation d’un graphe sous la forme d’un type abstrait de donnée (TAD).

Pour cela nous avons tout d’abord développé plusieurs classes qui ont permis de modéliser les différents constituants d’un graphe, à savoir :

-la classe Arc caractérisée par ses attributs d’instance *destination* et *cout*

-l’interface Graphe (que nous avons par la suite décidé d’altérer afin de faire intervenir un labyrinthe qui correspond à un graphe un peu spécial).

-la classe Arcs qui est simplement une liste contenant plusieurs objets de type Arc.

-la classe GrapheListe qui implémente l’interface Graphe, permet de gérer un graphe en lui ajoutant des sommets (nœuds), arêtes (arcs) et une pondération (cout).

-la classe Main où est instancié un objet de type GrapheListe et qui permet d’afficher le graphe exemple de la figure 2 du sujet.

Dans cette partie, les tests ont uniquement consisté en une vérification de la bonne construction d’un graphe

Partie 2 – Calcul du plus court chemin par point fixe

Algorithme de Bellman-Ford

Durant cette seconde partie nous avons dans un premier temps écrit l’algorithme de Bellman-Ford, puis nous l’avons testé dans le programme main de la classe Main précédemment créée.