\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\***TRAJET OPTIMAL**\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Programme C++

par Antoine Picard 13/02/2015

I) PRESENTATION:

L’objectif de ce projet est de proposer un programme qui détermine la distance minimale entre deux sommets d’un graphe orienté. Le programme proposé permet à l’utilisateur d’entrer dans le terminal de commande les deux extrémités du chemin qu’il cherche à minimiser, et retourne par la suite la distance minimale entre ces deux points ainsi que le trajet optimal emprunté pour parcourir la distance minimale.

L’algorithme de recherche de trajet optimal utilisé ici est l’algorithme de Dijkstra qui est présenté plus en détail en commentaire dans le fichier graph.cpp.

Le fichier de type ‘data.txt’ contenu dans le dossier data (qui est lui même dans le répertoire courant) est une représentation d’un graphe orienté.

Il est construit de la manière suivante :

- Chaque ligne représente une arête du graphe orienté.

- Chaque ligne est constituée de 3 entiers séparés par des espaces, avec :

- - 1er entier : Sommet initial de l’arête du graphe.

- - 2ème entier : Sommet terminal de l’arête du graphe.

- - 3ème entier : Distance séparant le sommet initial et le sommet terminal.

Cette note de présentation/utilisation sera brève étant donné que les différents fichiers sources sont tous commentés quasiment ligne par ligne. Pour des raisons indépendantes de ma volonté le programme ne peut-être ouvert avec Visual Studio (le travail a été fait intégralement sous Mac), on donne donc la marche à suivre (très simple) pour exécuter le programme sans erreur.

II) UTILISATION:

La marche à suivre pour exécuter le programme utilise le terminal sous Mac (équivalent de l’invité de commandes sous Windows).

On doit dans un premier temps se placer dans le dossier contenant le projet.

On utilise ensuite la commande suivante pour compiler le code:

> g++ arete.cpp graph.cpp main.cpp

Cette commande entraine la création d’un fichier exécutable ‘a.out’

On utilise la commande suivante pour exécuter le programme:

> ./a.out (sous Mac)

> a (sous Windows)

Le terminal demande ensuite à l’utilisateur d’entrer le point d’origine ainsi que le point destination.

Il affiche ensuite la distance minimale entre ces deux points et le trajet optimal emprunté.

On peut évidemment changer le graphe que l’on souhaite utiliser, pour cela on a deux possibilités:

1) Modifier dans le fichier ‘main.cpp’ la commande :

> graph G("data/data.txt");

En précisant le chemin du nouveau fichier à utiliser.

2) Placer le fichier que l’on souhaite utiliser pour la création du nouveau graphe dans le dossier data, puis le nommer ‘data.txt’.