TD – Graphes

2ème partie

Introduction et objectifs

Dans cette deuxième partie du TD consacré aux graphes, l'objectif principal est la mise en œuvre de l'algorithme de Dijkstra pour un graphe connexe pondéré de poids strictement positifs.

On utilisera les tableaux (objets `array`) de la bibliothèque `numpy`.

Autour de l'algorithme de Dijkstra

Exercice N°1 – Mise en œuvre de l'algorithme

Le but de cet exercice est de construire un script complet comportant les parties principales suivantes :

1. Données (le graphe, un sommet de départ et un sommet d'arrivée).
2. Construction de la matrice des poids.
3. Mise en œuvre de l'algorithme de Dijkstra.
4. Affichage d'une chaîne de longueur minimale et de la longueur de cette chaîne.

La construction de la matrice des poids devra tenir compte du fait que le graphe peut, ou non, être orienté (on introduira un troisième argument booléen).

Pour la troisième partie, on appellera une fonction (préalablement écrite) Python `fDijkstra` qui :

- recevra en argument : la matrice des poids du graphe considéré, le numéro du sommet de départ et le numéro du sommet d'arrivée.
- renverra une variable `S` :
    - Valant `False` si aucune chaîne n'a été trouvée entre le sommet de départ et celui d'arrivée (cela peut se produire dans le cas d'un graphe orienté).
    - Correspondant sinon à une liste de listes à 3 éléments (il y en a autant que de sommets) telle que vue en cours.

L'affichage d'une chaîne de longueur minimale et de sa longueur s'effectuera dans le script en dehors de la fonction `fDijkstra` que l'on réutilise dans l'exercice suivant.

Exercice N°2 – Matrice des longueurs des chaînes optimales

On a souligné dans le cours que l'utilisation de l'algorithme de Dijkstra pour la détermination d'une chaîne optimale (i.e. de longueur minimale) entre les sommets de départ *sd* et d'arrivée *sa* fournissait en fait, dans le cas d'un graphe non orienté, les longueurs minimales entre *sa* et tous les autres sommets du graphe.

En vous servant de la fonction `fDijkstra` de l'exercice précédent, construire un script qui fournira la matrice symétrique `LO` des longueurs minimales entre les sommets.

Par exemple, avec le graphe du cours, on obtient le tableau :

```
[[ 0  14   5   3  15  13  26]
 [14   0   9  12  15  17  29]
 [ 5   9   0   8  10   8  21]
 [ 3  12   8   0  18  16  29]
 [15  15  10  18   0  18  14]
 [13  17   8  16  18   0  13]
 [26  29  21  29  14  13   0]]
```