Des graphes (E3A 2017)

Dans cet exercice, les programmes seront écrits avec le langage Python. On pourra utiliser si besoin la bibliothèque `numpy` :

- `zeros([n,p])` renvoie un tableau bidimensionnel (n, p) rempli de 0 ;
- si T est un tableau bidimensionnel, `T[i,j]` accède à l'élément ligne i et colonne j de ce tableau.

Le réseau de métro d'une grande ville comporte N stations réparties sur un certain nombre de lignes interconnectées (par exemple, pour la ville de Lyon, on a $N = 40$ avec 4 lignes).

On représente de réseau par un graphe non orienté $G = (\llbracket 0, N-1 \rrbracket, A)$ défini par :

- l'ensemble des sommets est $\llbracket 0, N-1 \rrbracket$; on a numéroté les stations de 0 à $N-1$. Le sommet i du graphe G représente la station i ;
- A désigne l'ensemble des arêtes de G. Une arête entre les sommets i et j (éléments de $\llbracket 0, N-1 \rrbracket$) n'existe que si les stations i et j sont des stations adjacentes sur une même ligne de métro. Ce graphe est représenté par la liste `Liste_A` (donnée en variable globale) de ses arêtes sous la forme `[i,j]` tels que $i < j$.

1. Écrire une fonction `voisin_G` qui prend en entrée un entier $i \in \llbracket 0, N-1 \rrbracket$ et retourne la liste des sommets adjacents à i dans G.

On suppose le graphe G connexe et on cherche à déterminer le trajet le plus rapide entre deux stations du réseau.

Soit `duree` la fonction définie sur A qui attribue à une arête (i, j) la durée du trajet entre la stations i et la station j en minutes, arrondie sur un nombre entier.

Un trajet entre deux stations i et j correspond à un chemin dans G qui part de i et arrive en j. La durée d'un tel trajet $i = i_0 \to i_1 \to \cdots \to i_n = j$ utilisant n arêtes est la somme des durées des arêtes :

$$\texttt{duree}(i = i_0 \to i_1 \to \cdots \to i_n = j) = \sum_{k=0}^{n-1} \texttt{duree}(i_k, i_{k+1}).$$

Pour simplifier, on ne tient pas compte des durées correspondante aux changements de métro.

On dispose de la liste `Duree` des listes `[i,j,duree(i,j)]` pour $0 \leqslant i < j \leqslant N-1$ tels que $(i, j) \in A$. On note `D` la valeur `D = sum(U[2] for U in duree]`.

2. Soient i, j deux sommets de G. Démontrer qu'il existe au moins un trajet de durée minimale entre i et j.
   Pour i, j sommets de G, on note $\delta_{\min}(i, j)$ la durée minimale d'un trajet entre i et j.
3. Soient i, j deux sommets de G. Soit $i = i_0 \to i_1 \to \cdots \to i_n = j$ un chemin dans G qui pqrt de i et arrive en j en utilisant n arêtes ($n \geqslant 1$). On suppose que ce chemin réalise un trajet de durée minimale enre i et j. Justifier que pour tout k entre 1 et n, $i = i_0 \to i_1 \to \cdots \to i_k$ est un trajet de durée minimale entre i_0 et i_k et $i_k \to \cdots \to i_n = j$ est un trajet de durée minimale entre i_k et j.
4. En déduire, pour i et j distincts tels que $(i, j) \notin A$, une expression de $\delta_{\min}(i, j)$ en fonction des valeurs $\delta_{\min}(i, k)$ et $\delta_{\min}(k, j)$ pour k parcourant la liste des sommtes de G à l'exception de i et de j. Justifier votre réponse.

5. Définir `Tinit` le tableau bidimensionnel (N, N) tel que

$$\forall (i,j) \in [\![0, N-1]\!]^2,\ \texttt{Tinit[i,j]} = \begin{cases} 0 & \text{si } i = j \\ \texttt{duree(i,j)} & \text{si } (i,j) \in A \\ D & \text{sinon} \end{cases}.$$

6. Définir la fonction `FW` qui prend en entrée un tableau bidimensionnel (N, N) `T` et retourne le tableau bidimensionnel (N, N) `T'` défini par

$$\forall (i,j) \in [\![0, N-1]\!]^2,\ \texttt{T'[i,j]} = \min(\texttt{T[i,j]}, \texttt{T[i,k]+T[k,j]}, k \in [\![0, N-1]\!]).$$

7. Écrire un programme qui, en utilisant le tableau `Tinit` et la fonction `FW`, permet de calculer un tableau bidimensionnel (N, N) dont le (i,j)-ième coefficient vaut $\delta_{\min}(i,j)$, pour tous sommets i et j. Combien d'itérations sont-elles nécessaires pour conclure ?