GRAPHES ET RÉSEAUX **ISC - HEIG-VD**
J.-F. Hêche **Printemps 2022**

Travail pratique **Algorithme de Dijkstra bidirectionnel**

Objectifs

Dans ce travail pratique vous devez programmer et comparer l'algorithme de Dijkstra et sa variante bidirectionnelle pour le calcul d'un plus court chemin entre deux sommets donnés d'un réseau.

Données et sources fournies

Les réseaux utilisés dans ce travail sont stockés dans des fichiers texte dont la première ligne contient le nombre n de sommets du graphe. Les n lignes suivantes contiennent chacune le numéro d'un sommet suivi de ses deux coordonnées cartésiennes. Les sommets sont numérotés de 0 à $n-1$ et apparaissent dans l'ordre croissant. Finalement les n dernières lignes contiennent, pour chaque sommet dans l'ordre croissant, son numéro suivi de la liste des numéros de ses voisins.

Les sources fournies vous permettent de lire, stocker et manipuler un réseau stocké dans un fichier respectant le format précédent. Même si les graphes sont traités comme des graphes orientés, tous les réseaux fournis sont, en fait, non orientés (ou « symétriques ») et il n'y a donc pas de différences entre la liste des successeurs d'un sommet et celle de ses prédécesseurs.

Finalement le poids d'un arc correspond à la distance euclidienne entre ses deux extrémités, distance arrondie à l'entier le plus proche.

Travail de programmation à effectuer

Vous devez compléter les sources fournies afin d'obtenir une mise en œuvre correcte et efficace de l'algorithme de Dijkstra pour le calcul d'un plus court chemin d'une origine s à une destination t ainsi que la variante bidirectionnelle de l'algorithme. Pour cette dernière votre code devra alterner strictement les itérations en avant et celles en arrière.

Vos méthodes devront évidemment fournir la distance entre s et t mais également l'itinéraire optimal calculé ainsi que le nombre de sommets traités (pour la comparaison des approches).

Vous placerez tout votre code dans un package (et d'éventuels sous-packages) dont le nom respectera le format `nom.prenom`.

Travail d'analyse à effectuer

Pour étudier l'intérêt de la variante bidirectionnelle vous comparerez le nombre de sommets traités pour chaque algorithme lors du calcul d'un plus court chemin. Plus spécifique-

ment vous générerez aléatoirement 1000 couples (s, t) de sommets du réseau stocké dans le fichier `R10000_1.txt`. Pour chaque couple vous déterminerez le nombre de sommets traités par les deux variantes de l'algorithme. Vous analyserez et commenterez ensuite les résultats obtenus.

Pour choisir les indices des sommets, vous utiliserez le générateur de nombres pseudo-aléatoires fournie par la classe `Random` (du package `java.util`) et plus spécifiquement la méthode `nextInt(int)` de cette classe. Vous initialiserez votre générateur avec la graine (*seed*) 20220404.

Modalités et délais

- ▷ Le travail de programmation est à effectuer en Java, version 11.
- ▷ Les sources à compléter et les fichiers de données sont disponibles sur le site Cyberlearn du cours.
- ▷ Vous devez rendre les sources complètes, soigneusement documentées, de votre travail ainsi qu'un court rapport, au format PDF, contenant l'analyse statistique commentée des résultats obtenus. Vous réunirez tous vos documents dans une archive au format `zip`.
- ▷ Vous devez rendre votre travail sur Cyberlearn au plus tard le **lundi 9 mai 2022** (avant minuit).