

Algorithmique et structures de données 2

TP5

am@up8.edu

2022-2023

1 Consignes de rendu

TP à rendre pour le lundi 17/04 12h (deadline fixe)

Un projet qui ne compile pas, qui termine sur une erreur de segmentation ou qui ne respecte pas les consignes ci-dessous ne sera pas corrigé.

- fichiers à rendre :
 - au minimum, fichiers `dijkstra.h` et `dijkstra.c` où vous définissez et implémentez les fonctions dont vous avez besoin pour calculer les distances entre un noeud et les autres noeuds du graphe selon l'algorithme de Dijkstra.
 - fichier `tp5.c` où vous testez les fonctions du module ci-dessus.
 - Makefile (cibles `all`, `clean`, `dist`)
- dépôt : archive `tp5_votre_nom.zip`
- vous pouvez ajouter un fichier `README.txt` au besoin (ne m'envoyez pas d'information complémentaire par mail / *via* mattermost)
- vous pouvez rendre le TP seul.e ou à deux. Dans ce cas, déposez la même archive avec vos deux noms sur vos deux moodles respectifs.

2 Algorithme de Dijkstra

Commencez par déclarer une taille de graphe (correspondant au nombre de noeuds) et un tableau de tableaux correspondant à la matrice d'adjacence du scenario 1 vu en cours. Les valeurs ∞ sont remplacées par la valeur 9999 (que vous pouvez stocker dans une variable globale `INFINITY`).

Une matrice :

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

est définie comme suit :

```
1 int matrice = {{1,2},  
2 {3,4}  
3 }
```

2.1 Question 1 :

Reprenez les étapes vues en cours, demandant la saisie au clavier du noeud de départ et permettant d'obtenir l'affichage suivant (créez une fonction affichage dédiée) :

Le nombre de noeuds a été fixé dans une variable globale MAX = 7
Entrez le numéro du noeud de départ : 1

Distance entre noeud 0 et noeud 2 = 12
Chemin à parcourir = 2<-6<-0

Distance entre noeud 0 et noeud 3 = 14
Chemin à parcourir = 3<-1<-6<-0

Distance entre noeud 0 et noeud 4 = 27
Chemin à parcourir = 4<-2<-1<-6<-0

Distance entre noeud 0 et noeud 5 = 7
Chemin à parcourir = 5<-6<-0

Distance entre noeud 0 et noeud 6 = 12
Chemin à parcourir = 6<-0

Distance entre noeud 0 et noeud 7 = 5
Chemin à parcourir = 7<-0

Vous aurez besoin d'initialiser puis de mettre à jour :

- un **tableau indiquant pour chaque noeud si il a été visité ou non**,
- un **tableau indiquant la distance entre le noeud de départ et chaque noeud du graphe** (ce tableau est mis à jour au fur et à mesure),
- un **tableau de prédécesseurs** donnant pour chaque noeud le numéro du noeud le précédent dans le chemin le plus court identifié.

2.2 Question 2 :

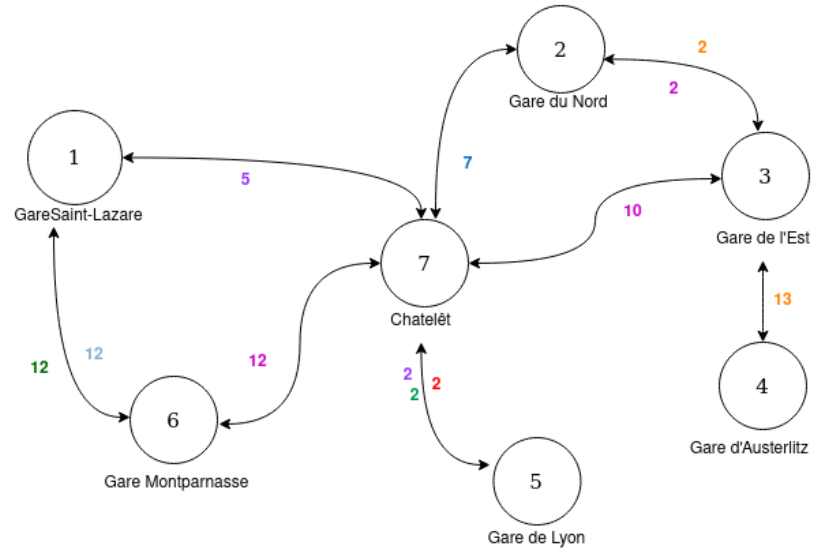
Dans le fichier tp5.c, faites les appels nécessaires pour afficher :

1. les distances entre Gare du Nord et toutes les autres gares dans le scénario 1 du cours;
2. les distances entre Gare du Nord et toutes les autres gares dans le scénario 2 du cours.

2.3 Question 3 :

Implémentez une seconde fonction `affichage2` affichant les chemins "dans le bon sens" (de gauche à droite).

Graphe orienté pondéré



**Graphe orienté pondéré ligne 4 ne circule pas dans le sens
Porte de Clignancourt vers Bagneux**

