

TP - Bases

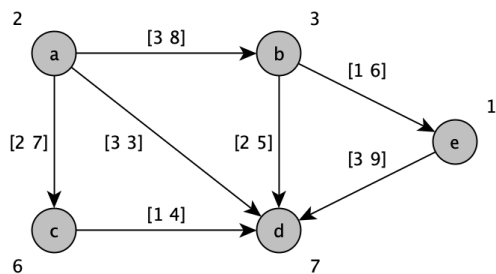
Ce TP est à réaliser par groupe de **2 étudiants maximum**.

A rendre en fin de séance sur Moodle

Objectif : implémenter une structure de graphe ainsi que quelques algorithmes de base.

Vous devez :

1. Implémenter une **structure de graphe simple**.
2. Créer une fonction d'**importation d'un graphe** (orienté ou non) et afficher le contenu du graphe (pas de manière graphique). Des graphes sont fournis sur Moodle (voir au verso de cette feuille).
3. Ajouter à la structure de données un tableau des **degrés des sommets** (dans le cas orienté et non-orienté) à créer au moment de l'importation du graphe.
4. Implémenter l'algorithme **PlusCourtCheminNbArcs** (dernier Slide de la partie « Parcours dans les graphes ») qui est une adaptation du parcours en largeur pour calculer le plus court chemin, en termes de nombre d'arcs (on ne tient pas compte ici des éventuelles valeurs sur les arcs), du sommet passé en paramètre à tous les autres sommets du graphe.
5. Modifier l'algorithme précédent afin qu'il permette de retrouver les plus courts chemins.
6. Modifier l'algorithme précédent afin qu'il renvoie le résultat uniquement entre un sommet initial passé en paramètre et un sommet terminal passé également en paramètre.
7. Tester l'ensemble de vos algorithmes sur les graphes fournis sur Moodle
8. Déposer votre code sur Moodle à la fin de la séance (je dois pouvoir l'exécuter facilement), ainsi que vos résultats dans un petit document PDF.

cours-representation.gra

```

NAME: cours-representation.gra
COMMENT: graphe du cours ...
ORIENTED: true
NB_VERTICES: 5
NB_VALUES_BY_VERTEX: 1
NB_EDGES: 7
NB_VALUES_BY_EDGE: 2

```

```

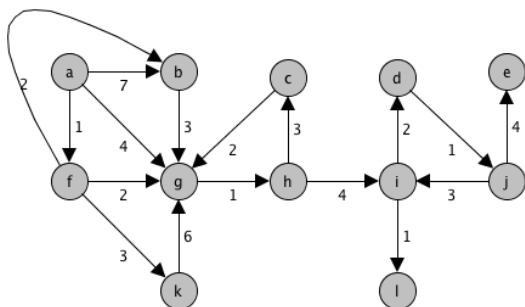
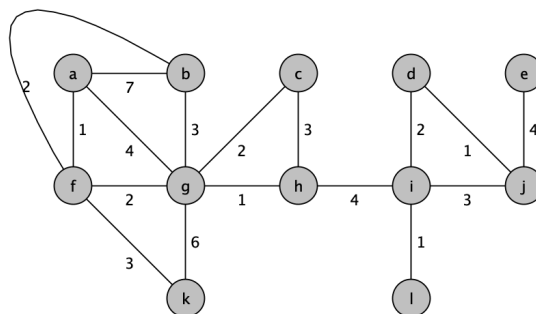
VERTICES [id name values[]]:
0 a 2
1 b 3
2 c 6
3 d 7
4 e 1

```

```

EDGES [idInitialVertex idFinalVertex values[]]:
0 1 3 8
0 2 2 7
0 3 3 3
1 3 2 5
1 4 1 6
2 3 1 4
4 3 3 9

```

graphe-oriente-01.gra**graphe-nonoriente-01.gra****graphe-communes.gra**

Graphe de voisinage des communes de France