Bibliothèque de manipulation de graphes en C

Présentation générale

La bibliothèque permet de manipuler des graphes orientés ou non. Les graphes sont toujours valués, et le type des valeurs des arcs/arêtes est double (flottant double précision). La valeur nulle pour un arc/arête est interdite.

L'intérêt de cette bibliothèque est de pouvoir charger un fichier texte ou générer aléatoirement des graphes, et de pouvoir consulter avec des fonctions d'interface des informations comme les voisins, successeurs, ...

En interne, les données sont stockées pour améliorer l'efficacité des opérations. La méthode est simple : les données sont stockés sous différents formats (liste de voisins, liste de successeurs, matrice d'incidence, ...) afin d'avoir des opérations de consultation en coût constant.

Les sommets des graphes sont des chaînes de caractères. Les sommets sont manipulés par leur numéro.

Installation/Compilation

Il n'y a rien à installer. Nous vous fournissons deux fichiers graphe.h et graphe.c

Vous utiliserez la compilation séparée. Télécharger le fichier d'exemple testAnalyseur.c et lancer la commande

```
$ gcc -Wall -g -std=c99 testAnalyseur.c graphe.c -o testAnalyseur
```

Votre programme se lance avec

```
 $ ./testAnalyseur <fichier_graphe>
```

où fichier_graphe est un fichier contenant un graphe au format décrit dans la section suivante.

Pour tester la commande précédente, télécharger le fichier ex-no.grp et lancez

```
 $ ./testAnalyseur ex-no.grp
```

qui devrait vous afficher le contenu des graphe dans une version détaillée

Syntaxe des fichiers de graphe

Vous pouvez mettre des commentaires d'une ligne grâce au caractère # suivi du commentaire. Le # peut apparaître n'importe où dans la ligne.

Le fichier doit avoir la syntaxe suivante :

```
type = <oriente OU non-oriente>;

sommets = <liste de sommets séparés par des espaces> ;

arcs = <liste d'arcs séparés par des espaces, un arc étant représenté par
Origine Destination Valeur_de_l'arc> ;
```

Les arcs ont par défaut le type double (flottant double précision).

Exemple d'un graphe non orienté :

```
# type
type = non-oriente;

# la liste des sommets
sommets = A B C D E;

# la liste des arcs
arcs =
     A  B 10
     B  C 20
     A  E 15
;
```

Exemple d'un graphe orienté :

```
# type
type = oriente;

# la liste des sommets
sommets = bleu vert jaune rouge;

# la liste des arcs
arcs =
     bleu  jaune 1.5
     jaune rouge 2.2
     jaune vert 5.5
;
```

Consignes d'utilisation de la bibliothèque

Les structures de données sont commentées, mais uniquement dans un but pédagogique. Lorsque vous utilisez la bibliothèque, vous ne devez **jamais** accèder directement aux structures. Vous devez utiliser les fonctions mises à votre disposition.

L'intérêt est que si les structures de données de la bibliothèque changent en cours d'année, votre code continuera de fonctionner. De plus, lors du contrôle TP de fin de semestre, vous pourriez avoir une version de la librairie avec des structures de données différentes, il vaut mieux donc apprendre à utiliser uniquement les interfaces de fonctions.

Si vous estimez que des fonctions manquent, vous pouvez toujours en parler à votre enseignant et le convaincre de la rajouter.

Vous devez toujours utiliser l'option `-Wall`. En C, un avertissement est très souvent une erreur. Vous êtes encouragés à utiliser en plus les options `-g -std=c99`.