

Consignes

Les consignes de la première fiche de TP restent valides.

Attention : vous devez déposer votre travail à la fin des deux séances consacrées à cette fiche sur Moodle <https://moodle.uphf.fr/course/view.php?id=1681>

1. Le travail en binôme est autorisé.
2. N'oubliez pas d'indiquer vos nom(s) et prénom(s) en commentaire en haut de votre (vos) fichier(s).
3. Ne déposez que les fichiers sources (code) de votre travail.
4. Ne négligez pas les commentaires pour expliquer vos choix et les fonctions mises en œuvre.

Le respect des consignes sera pris en compte dans la notation.

La version déposée à la fin de la séance sera celle évaluée. Néanmoins une version améliorée / complétée pourra être déposée au plus tard une semaine après la séance, et pourra être prise en compte sous la forme d'un *bonus* dans la notation.

Exercice 1 : représentation par matrice d'adjacence

Cet exercice est similaire à l'exercice 1 de la première fiche, mais avec l'objectif de manipuler un graphe via une matrice d'adjacence.

On associe cette fois à un graphe les éléments de base suivants : un type (orienté ou non orienté) ; le nombre n de sommets ; le nombre m de connexions (arcs ou arêtes) ; une matrice (carrée) d'adjacence pour représenter les connexions entre les sommets.

Ici encore les sommets sont caractérisés par un identifiant unique géré, pour simplifier, par un entier (cf fiche 1).

Après avoir défini une structure de données appelée **Matrice** (en ajoutant éventuellement d'autres données que vous jugez utiles), vous écrivez des fonctions pour :

- Créer un graphe pour un type donné et un nombre de sommets donné. La fonction initialisera le graphe **sans** connexion (pas d'arcs / arêtes) mais devra saisir (et stocker) les identifiants des sommets.
- Libérer l'espace mémoire associé à un graphe donné (selon le langage utilisé).
- Ajouter une connexion dans un graphe entre deux sommets connus par leurs identifiants.
- Supprimer une connexion dans un graphe entre deux sommets connus par leurs identifiants.
- Ajouter un sommet dont l'identifiant est passé en paramètre dans un graphe. Ce sommet est initialement sans connexion. La fonction n'ajoutera pas le sommet si l'identifiant existe déjà.
- Savoir si un sommet connu par son identifiant est un sommet adjacent **direct** d'un autre sommet donné.
- Afficher un graphe. On se contentera ici d'afficher le type du graphe, le nombre de sommets puis la matrice d'adjacence (en ligne).
- Charger un graphe à partir d'un fichier **texte** dont le nom physique est passé en argument et respectant **scrupuleusement** le format suivant : la première ligne contient uniquement le type du graphe (0 pour non orienté, 1 pour orienté), le nombre de sommets (n) et le nombre de connexions (m). Puis on a une ligne pour chaque connexion ($\rightarrow m$ lignes consécutives), contenant les deux identifiants des extrémités de l'arc ou de l'arête, séparés par un espace. Ainsi par exemple on pourrait avoir un fichier intitulé "mon_graphe.txt" et contenant :

```
1 7 6
1 2
1 3
```

3 4
5 1
5 6
4 7

- Sauvegarder un graphe dans un fichier texte dont le nom physique est passé en argument, et en respectant **scrupuleusement** le format de la question précédente.

Exercice 2 : parcours

On considère le problème défini comme suit. Soit en entrée un mot de départ, un mot de fin, et un ensemble de mots (i.e., un lexique), tous ayant la même longueur. Trouver une séquence de transformations la plus courte possible pour passer du mot de départ au mot final de telle sorte qu’une lettre (exactement) soit modifiée à chaque étape de la séquence, et que chaque mot soit dans le tableau. Si aucune séquence n’existe la méthode retournera *null* (ou un équivalent au choix selon le langage de programmation utilisé pour indiquer qu’il n’y a pas de solution). Sinon la méthode permettra d’afficher la séquence des transformations.

Exemple :

Supposons que le tableau de mots soit CLOT - FLAN - FLOT - ILOT - PLAN - PLAT - PLOT - VLAN et que l’on veuille partir de ILOT pour arriver à VLAN alors on peut trouver la séquence ILOT - PLOT - PLAT - PLAN - VLAN.

Si on utilise le tableau CLOT - FLAN - ILOT - PLAT - PLOT - VLAN alors il n’est pas possible de partir de ILOT et d’arriver à VLAN.

Nous pouvons modéliser ce problème sous la forme d’un graphe non orienté avec un sommet pour chaque mot du tableau, et une arête entre deux sommets si et seulement si la modification d’un caractère exactement permet de passer d’un mot à l’autre. Une fois le graphe construit le problème peut être résolu en appliquant l’algorithme de parcours en largeur afin de trouver un plus court chemin (en nombre d’arêtes) entre les deux sommets associés aux mots de départ et d’arrivée.

Implémenter les algorithmes nécessaires pour résoudre ce problème, en représentant le graphe via sa matrice d’adjacence ou ses listes d’adjacence, au choix. Le tableau de mots sera fourni via un fichier texte contenant sur la première ligne le nombre de mots, suivi de chaque mot (1 par ligne). Il n’est pas garanti que les mots soient dans l’ordre lexicographique dans le fichier.

Les mots “origine” et “destination” seront fournis en argument du programme.