

L'objectif de cette première fiche est d'implémenter les structures de données et les fonctionnalités de base pour manipuler un graphe, **orienté ou non**. Ces fonctionnalités serviront à la conception et l'implémentation des algorithmes des fiches suivantes, il est donc important de traiter l'ensemble des questions.

Exercice 1 :

1. Première partie : représentation par matrice d'adjacence

On associe à un graphe les éléments de base suivants : un type (orienté ou non orienté) ; un nombre n de sommets ; un nombre m de connexions (arcs ou arêtes) ; une matrice carrée représentant les connexions, entre sommets. Les sommets sont caractérisés par un identifiant unique (une chaîne de caractères).

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

- Initialiser un graphe pour un type (orienté ou non) et un nombre de sommets donnés. La fonction initialisera le graphe sans arcs/arêtes.
- Libérer l'espace mémoire associé à un graphe donné.
- Ajouter une connexion entre deux sommets donnés.
- Supprimer une connexion entre deux sommets donnés.
- Ajouter un sommet dans le graphe. Ce sommet est initialement sans connexion.
- Savoir si un sommet donné est adjacent ou est un successeur direct d'un autre sommet donné.
- Calculer le degré (et les demi-degrés si le graphe est orienté) de l'ensemble des sommets.
- Charger un graphe à partir d'un fichier `texte` respectant scrupuleusement le format suivant : la première ligne contient le nombre de sommets, un espace, et le type du graphe (valeur 1 pour orienté, 0 pour non orienté). Puis, un arc ou une arête par ligne (soit m lignes consécutives), identifié par deux valeurs séparées par un espace et représentant les sommets source et destination (cas orienté) ou les deux sommets adjacents (cas non orienté).
- Sauvegarder un graphe dans un fichier texte en respectant le format de la question précédente.
- Afficher un graphe. On se contentera ici d'afficher le nombre de sommets, le type du graphe, puis la matrice (ligne par ligne) et les degrés des sommets.

2. Deuxième partie : représentation par listes d'adjacence

On associe ici à un graphe les éléments de base suivants : un type (orienté ou non orienté) ; un nombre n de sommets ; un nombre m de connexions (arcs ou arêtes) ; un tableau de listes représentant les sommets successeurs / adjacents de chaque sommet. Comme précédemment les sommets sont identifiés de façon unique par une chaîne de caractères.

Après avoir défini une structure de données `ListAdj` (en ajoutant éventuellement d'autres données que vous jugez utiles), vous écrivez des fonctions pour réaliser les mêmes opérations que dans la première partie.

Exercice 2 :

Écrire des fonctions permettant de passer d'une représentation donnée à l'autre.