

ALGO5 : Graphes

Représenter un réseau social

Nous nous intéressons ici à une représentation simple de relations entre des personnes, comme dans un réseau social. Ces relations forment un graphe orienté que vous allez représenter comme tel et sur lesquels vous pourrez expérimenter les algorithmes de parcours des graphes.

Les fichiers `reseau1.txt` et `reseau2.txt` contiennent des informations qui proviennent de deux réseaux sociaux différents, mais ils respectent les mêmes règles syntaxiques. Ils contiennent d'abord la liste des personnes du réseau, puis une liste de villes. Ensuite, une description détaillée de chaque personne est donnée : la liste de ses personnes suivies, et la ville où la personne habite. On supposera que toutes les villes dans lesquelles une personne du réseau habite sont présentes dans la liste des villes précédentes. La dernière donnée du fichier est la liste des routes directes connues entre les villes et la distance kilométrique. Vous allez construire deux structures de données distinctes : une pour les personnes et ses abonnements (personnes suivies), une autre pour les villes et les routes.

Définissez les types et les variables nécessaires pour représenter les éléments suivants :

- un tableau `villes` de chaînes de caractères
- une matrice `distance_villes` qui contient des entiers (c'est une matrice carrée de la même dimension que le tableau `villes`)
- une structure `Personne` qui contient :
 - un champ `nom` de type chaîne de caractères
 - un champ `ville` de type chaîne de caractères
 - un champ `abonnements` de type liste chaînée de `Personne`
- un tableau `tab_personnes` de pointeurs sur des `Personne`

Ensuite, munissez-vous de quelques fonctions pour modifier votre réseau :

- écrire une fonction `ajoute_personne` qui prend un nom et une ville en paramètres. Cette fonction crée une nouvelle personne (sans abonnements) et l'ajoute à `tab_personnes`.

- écrire une fonction `get_personne` qui prend un nom en paramètre et retourne un pointeur vers cette personne en cherchant dans `tab_personnes`.
- écrire une fonction `ajoute_abo` qui prend deux personnes `p1` et `p2` en paramètres et ajoute `p2` aux abonnements de `p1`.
- écrire une fonction `enleve_abo` qui prend deux personnes `p1` et `p2` en paramètres et enlève `p2` aux abonnements de `p1`.
- écrire une fonction qui permet d'afficher une personne : son nom, sa ville, le nom des personnes qu'il suit.

Enfin, récupérez les informations données dans l'un des deux fichiers de données :

- écrire une fonction `parsePersonnesInfos` qui prend la liste de personnes (exemple : `[kevin,karim,alice,paul,charlotte,gabriel,dalila],`) et crée puis commence à remplir le tableau `personneTab`.
- écrire une fonction `parseVillesInfos` qui prend la liste de villes (exemple : `[lille,lens,arras,douai,carvin,seclin],`) et crée puis remplit le tableau `villes`.
- écrire une fonction `parsePersonne` qui prend une chaîne en entrée (de type : `kevin: {[gabriel,dalila],lille},`) et qui ajoute les infos de la personne ainsi que ses abonnements.
- écrire une fonction `parseVille` qui prend une chaîne en entrée (de type : `(lille,22,lens),`) qui met à jour la matrice `distancesVilles`.

Parser l'un des deux fichiers disponibles sur moodle pour tester le remplissage de vos structures.

Naviguer dans le réseau

On veut maintenant connaître les chaînes d'abonnements à travers le réseau, et en particulier la distance entre deux personnes du réseau : chaque personne est à une distance 0 d'elle-même; si une personne est abonnée directement à une autre, elle est à une distance de 1; etc.

Si on cherche à calculer la distance d'une personne A à une personne B, l'idée est de faire un parcours en largeur d'abord du graphe des abonnements à partir de A. On va créer un tableau de la dimension de `tab_personnes` dans lequel on va écrire, au fur et à mesure les distances de A à chacune des personnes du réseau.

Au départ, ce tableau contient 0 pour A, et -1 pour toutes les autres personnes. Puis on va regarder les abonnements de A : ils sont tous à une distance 1 de A. On regarde ensuite les abonnements des abonnements, etc etc. Question : quand allez-vous vous arrêter?

Écrire une fonction `distance_personnes` qui prend deux personnes `p1` et `p2` en paramètres et calcule la distance de `p1` à `p2`.

Écrire une fonction `distance_moyenne` qui calcule la distance moyenne entre deux personnes dans un réseau (on ne prendra pas en compte dans le calcul les personnes qui ne sont pas en lien).