

Travaux dirigés n° 1

Graphes - Représentation

Exercice 1 (Preliminaire - Listes chaînées)

Une liste doublement chaînée l comporte un attribut *tete* qui pointe sur le premier élément de la liste. Dans une telle liste, chaque élément est un objet comportant un attribut *cle* et deux autres attributs pointeurs : *succ* et *pred*.

1°) Illustrez les trois états successifs suivants :

1. Une liste représentant l'ensemble 9, 16, 4, 1 ;
2. La liste après insertion, en tête de liste, d'un élément dont la clé vaut 25 ;
3. La liste après suppression de l'élément dont la clé vaut 4.

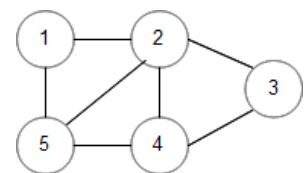
2°) Définissez les procédures de liste suivantes :

- a) **rechercher-liste** : trouve le premier élément dont la valeur de la clé est spécifiée et retourne un pointeur sur cet élément. Si aucune élément ne possède cette valeur, la procédure retourne NIL ;
- b) **insérer-liste** : étant donné un élément dont l'attribut *cle* a déjà été initialisé, greffe l'élément en tête de liste.
- c) **supprimer-liste** : élimine un élément de la liste chaînée dont un pointeur lui est fourni.

Exercice 2 (Représentation des graphes)

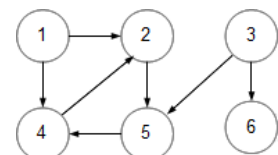
1°) Soit le graphe non orienté ci-contre, donnez

- a) sa taille ;
- b) son ordre ;
- c) sa représentation par listes d'adjacences ;
- d) sa représentation par matrice d'adjacences.



2°) Soit le graphe orienté ci-contre, donnez sa représentation

- a) par listes d'adjacences ;
- b) par matrice d'adjacences.



3°) En fonction de son nombre de sommets et de son nombre d'arêtes ou d'arcs, donnez la quantité de mémoire nécessaire pour stocker un graphe par

- a) listes d'adjacences ;
- b) matrice d'adjacences.

4°) Donnez le temps d'exécution nécessaire pour afficher tous les sommets adjacents à un sommet u lorsque le graphe est représenté par

- a) listes d'adjacences ;
- b) matrice d'adjacences.

5°) Donnez le temps d'exécution nécessaire pour déterminer si l'arête ou l'arc $(u, v) \in A$ lorsque le graphe est représenté par

- a) listes d'adjacences ;
- b) matrice d'adjacences.

Exercice 3 (Construction d'un graphe en langage C)

Le fichier texte `graphe.txt` ci-contre contient les données nécessaires à la représentation d'un graphe. Les 3 premières lignes contiennent chacune un couple chaîne de caractères-valeur permettant de spécifier les caractéristiques du graphe : le nombre de sommets (entier), l'état orienté ou non du graphe (booléen) et l'état valué ou non du graphe (booléen).

Ensuite, la définition des arêtes du graphe est délimitée par les indicateurs `DEBUT_DEF_ARETES` et `FIN_DEF_ARETES`. Entre ces indicateurs, on retrouve sur chaque ligne la définition d'une arête, c'est-à-dire deux entiers correspondant aux sommets de ses deux extrémités.

1°) Illustrez le graphe, les listes d'adjacences et la matrice d'adjacence correspondant aux données de ce fichier.

2°) Écrivez le code en langage C des fichiers d'entête `cellule.h` et `liste.h` permettant de déclarer une liste chaînée pouvant être utilisée pour représenter les listes d'adjacences pour ce type de graphe. Autrement dit, vous n'avez pas à écrire les fichiers `cellule.c` et `liste.c`, mais seulement la déclaration des structures de données et prototypes (signatures) des fonctions nécessaires à leur fonctionnement. Dans la suite, vous pouvez appeler ces fonctions si vous en avez besoin.

3°) Proposez, en langage C, une structure de données permettant de représenter un graphe semblable à celui défini dans le fichier `graphe.txt`.

4°) Proposez, en langage C, une fonction `initialiser_graphe` qui prend en paramètre le nom d'un fichier ayant la structure du fichier `graphe.txt` et qui construit le graphe associé en utilisant la structure de données de la question précédente et en construisant les listes d'adjacence et la matrice d'adjacence du graphe.

graphe.txt

```
n_sommets 6
orienté 0
valué 0
DEBUT_DEF_ARETES
0 1
0 4
0 5
1 2
1 5
2 3
2 5
4 5
FIN_DEF_ARETES
```

Références

Cormen T. H., Leiserson C. E., Rivest R. L. et Stein C., "Algorithmique" 3e édition, Dunod, 2010.