

# Second TP, graphes et parcours

Vincent Gripon

1 mars 2010

On considérera dans ce TP que les noeuds des graphes sont numérotés de 0 à  $n - 1$ , où  $n$  est le nombre de noeuds du graphe en question. Pour simplifier les choses, on assimilera les numéros des noeuds aux noeuds eux mêmes.

## 1 Structures de graphes

On se propose de représenter un graphe en mémoire sous deux formes différentes : une version matricielle où le graphe est représenté par une matrice  $M$  telle que  $M_{i,j}$  vaut `true` si et seulement s'il existe une connexion du noeud  $i$  au noeud  $j$  et une version tableau de listes où la  $i$ -ème case du tableau contient la liste des noeuds successeurs du noeud  $i$ .

### Question 1)

Quels sont les principaux avantages/inconvénients de ces deux représentations ? Comment adapter ces représentations au cas où le graphe est valué ?

Pour la suite du TP, il est demandé de choisir une des deux représentations comme représentation principale.

### Question 2)

Programmer une fonction qui permette de convertir la donnée d'un graphe sous la forme que vous n'avez pas choisie en la donnée du même graphe décrit selon la forme que vous avez choisie.

### Question 3)

Programmer une fonction qui teste si un graphe est orienté.

## 2 Parcours

On s'intéresse ici à la programmation des parcours qui visitent tous les noeuds d'un graphe une seule fois au plus.

### Question 4)

Programmer une fonction qui parcourt un graphe donné à partir d'un sommet donné en profondeur d'abord. La fonction rendra la liste des sommets parcourus dans l'ordre du parcours.

**Question 5)**

Programmer la fonction et la procédure `fifopush` et `fifopop` qui permettent respectivement d'ajouter un élément dans une file et de récupérer un élément dans cette même file.

**Question 6)**

En déduire un algorithme qui parcourt un graphe à partir d'un sommet donné en largeur d'abord. Le résultat de cet algorithme sera la liste des sommets visités dans l'ordre du parcours.