Problème Scientifique et informatique

1er livrable – Projet LivInParis

Paul Tournier, Alexandre Rollet, Florent Seydoux Classe C

Explications sur les graphes

i – Algorithmes classiques pour parcourir un graphe

En effectuant une recherche web, nous avons vu qu'il y a deux algorithmes classiques qui permettent de parcourir un graphe : le BFS (Breadth-First Search = parcours en largeur) et le DFS (Depth-first search = parcours en profondeur).

Le BFS cherche tous les chemins en partant d'un seul nœud, donc nous pouvons créer deux méthodes, la première est le BFS et la deuxième permet de faire le BFS plusieurs fois selon le nombre de sommets et de tout ajouter dans une liste qui contient l'entièreté des chemins en partant de tous les nœuds.

ii – Vérifier si un graphe contient des circuits

Nous avons testé si un graphe contient des circuits dans la méthode parcours en profondeur. Pour faire cela, nous rajoutons simplement un if qui regarde si pendant le parcours, la case par laquelle on passe a déjà été visitée. Si oui, c'est que nous avons fait une boucle/un cycle en parcourant le graphe.

iii – Vérifier si un graphe est connexe

Un moyen de vérifier si un graphe est connexe est d'utiliser sa matrice d'adjacence M. On considère que notre matrice possède n sommets. Il y a donc au maximum $n - 1$ chemins possibles dans le graphe qui ne repassent pas deux fois par le même sommet. Si l'on utilise les puissances de la matrice d'adjacence, on peut avoir une information sur le nombre de chemins d'une certaine longueur. En calculant M^{n-1}, on obtient donc une matrice qui nous donne le nombre de chemins possibles entre chaque sommet. Si une seule des cases de cette matrice est nulle, cela signifie que le graphe n'est pas connexe, car l'un des sommets n'a aucune connexion avec un autre sommet.

Dans notre exemple de 34 membres de l'association de karaté, on calcule donc M^{33} (car le plus long chemin qui ne repasse pas par un sommet est de longueur 33). Cette matrice contient plusieurs valeurs nulles, donc on en déduit que le graphe qui représente les relations entre les membres n'est pas connexe : certains membres n'ont aucun lien avec d'autres membres.

```
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 231   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 165 132   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 165 132 132   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  66  33  33  33   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  66  33  33  33  99   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  66  33  33  33  33  66   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  99  99  99  99  33  33  33   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  33  99  66  66  33  33  33  66   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  33  33   0  33   0   0   0  33  66   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  66  33  33  33  33  33  99  33  33   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0  33  33  33  33  33  33  33  33   0  33   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  33  66  66  33  33  33  33  66  33   0  33  33   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  99  99  99  99  33  33  33 132  99  66  33  33  66   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0  33   0   0   0   0   0  66  33   0   0   0  33   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0  33   0   0   0   0   0  66  33   0   0   0  33  66   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  66   0   0   0  33  33  33   0   0   0  33   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  33  33  66  66  33  33  33  66  33   0  33  33  33  66   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0  33   0   0   0   0   0  66  33   0   0   0  33  66  66   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  33  33  66  66  33  33  33  66  66  33  33  33  33  99  33  33   0  66  33   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0  33   0   0   0   0   0  66  33   0   0   0  33  66  66   0   0  66  33   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  33  33  66  66  33  33  33  66  33   0  33  33  33  66   0   0   0  66   0  66   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0  33   0   0   0   0   0  66  33   0   0   0  33  66  66   0   0  66  33  66   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0  66   0   0   0   0   0  66  33   0   0   0  33  66  66   0   0  66  33  66   0  66   0   0   0   0   0   0   0   0   0   0   0
  33   0  33   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  66   0   0   0   0   0   0   0   0   0   0
  33   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  33   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0  33  33   0   0   0  33  33  33   0   0  33  33  33   0  33  66   0   0   0   0   0   0   0   0   0   0
  33  33   0  33   0   0   0  33  66  66   0   0   0  66  33  33   0   0  33  33  33   0  33  33   0  66  33   0   0   0   0   0   0   0
  66  33   0  33   0   0   0  33  66  66   0   0   0  66  33  33   0   0  33  33  33   0  33  33  33  33  33  66   0   0   0   0   0   0
   0   0  33   0   0   0   0   0  66  33   0   0   0  33  66  66   0   0  66  33  66   0  66  66   0  33  33  66  33   0   0   0   0   0
  66   0  99  33   0   0   0  33  66  33   0   0   0  66  66  66   0  33  66  66  66  33  66  66   0   0  33  33  33  66   0   0   0   0
   0  33  99  33  33  33  33  33  99  33  33  33  33  66  66  66   0  33  66  66  66  33  66  99  33  33  33  66  33  66  66   0   0   0
  99  66  33  33   0   0   0  33  99  66   0   0   0  66  33  33   0   0  33  33  33   0  33  66  33  66  66  99  99  66  66  33   0   0
 132  99 198  33   0   0   0   0  66   0   0   0   0   0  33  33   0   0  33   0  33   0  33  99  66  66  33  33  33  99  66  66 330   0


Le graphe du réseau des membres de l'association n'est pas connexe.
```

La matrice d'adjacence à la puissance 33 : elle montre que le graphe n'est pas connexe.

iv – Représentation graphique du graphe en utilisant l'aide d'une IA

Nous avons utilisé les extensions `Bitmap` et `Drawing`. On initialise d'abord une image de dimensions 3000x3000 pixels. Ensuite lors de la création du graphe on peut choisir le rayon du cercle qui le représente. Nous l'avons agrandi pour assurer qu'il soit plus visible et que l'on voit mieux les arêtes. A partir de la matrice, le code dessine tous les liens et représente les sommets par des cercles. On peut choisir la couleur et les tailles des sommets et traits. L'image .png est sauvegardée dans le fichier du code et est modifiée à chaque fois que le code est exécuté.