

- TP 6. Fermeture transitive d'un graphe orienté. -

Le but de ce TP est de calculer la fermeture transitive d'un graphe orienté D , puis de l'utiliser afin de calculer les composantes fortement connexes de D .

Langage. Programme en c++. Votre programme pourra commencer par :

```
#include <cstdlib>
#include <iostream>
using namespace std;
const int n=6;
int adjacence[n][n]={{0,0,0,1,0,1}, //La matrice d'adjacence du graphe oriente D.
                    {1,0,1,1,0,0},
                    {0,0,0,1,0,0},
                    {0,0,0,0,1,1},
                    {0,0,1,0,0,1},
                    {0,0,1,0,0,0}};
int fermeture[n][n]; // La matrice de la fermeture transitive de D.
```

Vous trouverez une entête à votre algorithme à l'adresse : <http://www.lirmm.fr/~thomasse/cours/TP/tp6.txt>

- Exercice 1 - Initialisation.

Ecrire une fonction void `initfermeture()` qui initialise le tableau **fermeture** de telle sorte que **fermeture** $[i][j] = 1$ s'il existe un chemin orienté de longueur au plus 1 de i à j , et que **fermeture** $[i][j] = 0$ sinon.

- Exercice 2 - Fermeture transitive.

Ecrire une fonction void `fermeturetransitive()` qui calcule les entrées du tableau **fermeture** de telle sorte que **fermeture** $[i][j] = 1$ s'il existe un chemin orienté de i à j , et que **fermeture** $[i][j] = 0$ sinon. Vérifier que la fermeture calculée correspond bien au graphe D .

- Exercice 3 - Composantes fortement connexe.

Ecrire une fonction void `comportfortconnexe()` qui affiche les composantes fortement connexes du graphe D . Le résultat pourra être de la forme:
Les composantes fortement connexes sont : {1, 4}, {2}, {3, 5, 6}, {7}.

- Exercice 4 - Pour aller plus loin.

- Engendrer un graphe orienté aléatoire sur $n = 500$ sommets et 500 arcs, puis calculer les tailles de ses composantes fortement connexe. Que peut-on dire de ce graphe?
- Même question avec 500 sommets et 1000 arcs.
- Afficher les composantes fortement connexes de telle sorte que si une composante C apparaît avant C' , il n'existe pas d'arc de C' vers C .