

Grands Réseaux d'Interaction

TP n° 2 : Extraction de composantes fortement connexes

Règles Générales

Les règles énoncées en en-tête du TP1 restent valables pour tous les TPs, sauf indication contraire explicite.

Vous devez remettre ce TP sur Moodle avant le **13 octobre**.

Composantes connexes

Les composantes connexes d'un graphe G sont les sous-graphes de G maximaux dans lesquels chaque sommet est accessible à partir de n'importe quel autre sommet. L'ensemble de toutes les composantes connexes forment une partition de G .

Si l'on parle de composantes connexes pour un graphe non-orienté, on utilise le terme de *composantes fortement connexes* lorsqu'il s'agit d'un graphe orienté.

Définition : Une composante fortement connexe d'un graphe orienté $G_d = (V, A)$ est donc un sous-graphe maximal de G_d dans lequel il existe un chemin entre chaque couple de sommet. On notera que la notion *chaque couple* signifie que pour deux sommets distincts $u, v \in V$, un chemin relie u à v et un autre v à u .

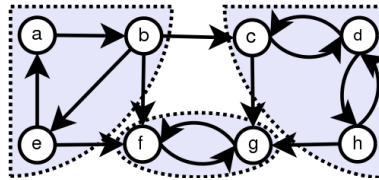


FIGURE 1 – Composantes fortement connectées d'un graph orienté (Wikimedia commons)

Travail demandé

Le travail demandé dans cette séance est de créer un programme permettant de calculer les composantes fortement connexes d'un graphe orienté. Pour cela, vous allez implémenter un algorithme basé sur des parcours de graphe, vus lors du TP1 : l'algorithme de Tarjan.

Algorithme de Tarjan

Cet algorithme a été proposé par Robert Tarjan en 1972. Il est de complexité linéaire par rapport au nombre de sommets *et* d'arcs du graphe, ce qui en fait un algorithme efficace.

Le principe est d'indexer les sommets les sommets visités au fur et à mesure de leur découverte. A chaque sommet $v \in V$ sont associées 2 valeurs : $index(v)$ et $decouverte(v)$. La valeur du plus grand $index$ de tous les sommets du graphe est notée I . On utilise également une pile P pour mémoriser les sommets scrutés.

Voici le principe de l'algorithme :

- Au départ, la pile est vide, et $I = 0$.

- On prend un sommet $u \in V$ pour lequel $index(u)$ n'est pas initialisé.
- On assigne $index(u) = decouverte(u) = I$.
 u est ajouté à la pile P . I est incrémenté.
- On répète récursivement l'étape suivante pour les successeurs de u non initialisés, jusqu'à ce qu'un successeur w d'un sommet scruté u' soit déjà dans la pile P .
- Dans ce cas, $index(u') = \min(decouverte(u'), index(w))$
- Lorsqu'on scrute un sommet v déjà dans P pour lequel $index(v) = decouverte(v)$, alors la découverte de la composante contenant v est terminée : tous les sommets empilés depuis v (inclus) forment une composante fortement connexe et sont dépilés.
- On reprend ensuite sur le dernier sommet présent dans P , ou si P est vide, sur un nouveau sommet non initialisé. Si il n'en reste plus, l'algorithme a terminé.

Vous trouverez un exemple de déroulement de l'algorithme sur

<http://bubuntu.net/graph/sessions/e-tarjan/exemple.html>

Il est important de bien comprendre le fonctionnement de l'algorithme avant de commencer l'implémentation !