

GRANDS RÉSEAUX D'INTERACTIONS

Masters SRI/SI M2 Université Paris 7

TP 4 (final) – Robustesse

Christophe Prieur et Clémence Magnien

prieur@liafa.jussieu.fr

clemence.magnien@lip6.fr

Attention, pour ce TP certains exercices ont deux versions différentes. Vous n'avez à faire que l'une des deux versions de chaque exercice, en fonction de la première lettre de votre nom de famille.

Les programmes de ce TP seront évalués pendant la soutenance. La date où vous devrez les rendre vous sera communiquée prochainement. Ils devront être accompagnés des programmes de soutenance, dont les spécifications sont en ligne.

Exercice 1 Écrire un programme qui, étant donné deux entiers n et m , construit un graphe aléatoire de n nœuds et m liens (en ajoutant m liens un à un dont les extrémités sont choisies au hasard).

On autorise le programme à créer des graphes qui ont des liens doubles (plus d'un lien entre deux mêmes nœuds).

Exercice 2 Comparer à l'aide de `gnuplot` les distributions de degrés des graphes donnés en début de cours à celles de graphes aléatoires de même nombre de nœuds et de liens.

Exercice 3 Écrire une fonction qui classe les nœuds d'un graphe dans un ordre aléatoire. Cette fonction retournera un tableau de taille n . La première case contiendra le numéro du premier nœud, la deuxième celui du deuxième nœud, et ainsi de suite.

Exercice 4 (Version 1, lettres de A à J)

On supprime les nœuds un à un en les choisissant au hasard. À l'aide des TP précédents, écrire un programme permettant de tracer l'évolution :

- du nombre de composantes connexes (TP1),
- de la taille de la plus grande composante connexe (TP1),
- de la distance moyenne (estimation, TP2) dans la plus grande composante connexe,
- du coefficient de clustering global du graphe (TP3),

en fonction du nombre de nœuds supprimés.

Le calcul des composantes connexes devra être fait par l'algorithme de connexité dynamique.

Attention : afin que le programme ne soit pas trop lent, on fera un certain nombre k de suppression avant de calculer un nouveau point...

Exercice 4 (Version 2, lettres de K à Z)

On supprime les nœuds un à un en les choisissant au hasard. À l'aide des TP précédents, écrire un programme permettant de tracer l'évolution :

- du nombre de composantes connexes (TP1),
- de la taille de la plus grande composante connexe (TP1),
- du diamètre (encadrement, TP2) dans la plus grande composante connexe,
- du coefficient de clustering local du graphe (TP3),

en fonction du nombre de nœuds supprimés.

Le calcul des composantes connexes devra être fait par l'algorithme de connexité dynamique.

Attention : afin que le programme ne soit pas trop lent, on fera un certain nombre k de suppression avant de calculer un nouveau point...

Exercice 5 Comparer à l'aide de `gnuplot` les courbes obtenues à l'exercice précédent sur les graphes donnés en début de cours à celles de graphes aléatoires de même nombre de nœuds et de liens.

Exercice 6 Écrire une fonction qui classe les nœuds d'un graphe par ordre de degrés décroissants.

Exercice 7 Refaire les exercices 4 et 5 mais cette fois-ci en supprimant les sommets par ordre de degrés décroissants : on supprime d'abord le sommet de plus fort degré, puis le second, etc.

Exercice 8 Comparer les résultats obtenus avec les deux stratégies de suppression.

Exercice 9 Avez vous d'autres idées de stratégies de suppression intéressantes ? Des idées pour rendre les graphes plus résistants ? ...