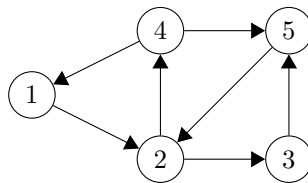


Descriptif du projet

Les réseaux d'interaction sont composés de nœuds (éléments) et liens qui relient les nœuds (interactions entre éléments). Ils sont devenus courants dans notre monde moderne : les réseaux d'infrastructure physique (autoroutes, réseau d'électricité, etc), les réseaux biologiques (interactions entre protéines), les réseaux sociaux (twitter, facebook), etc. Plusieurs bases proposant des réseaux issus de données réelles sont disponibles sur l'Internet :

- SNAP : <http://snap.stanford.edu/>
- KONECT : <http://konect.uni-koblenz.de/>
- Network Repository : <http://networkrepository.com/>

Les réseaux dans ces bases sont stockés dans des fichiers selon un format très simple : chaque fichier contient la liste des liens sous la forme d'un lien par ligne, constitué par ses deux nœuds extrémités (les nœuds sont numérotés de 1 à n et les lignes commençant par % sont des commentaires). Par exemple, le petit réseau ci-dessous :



peut être codé par le fichier suivant :

```

% un petit réseau de 5 noeuds
1 2
2 3
2 4
3 5
5 2
4 1
4 5
  
```

Le but de ce projet est de fournir un environnement pour pouvoir exploiter ces grands réseaux : exécution d'algorithmes sur ces réseaux, traitement par lots, sauvegarde et archivage des résultats, recherche de réseaux, etc.

Les réseaux considérés dans ce projet seront orientés ("Directed" en anglais, c'est à dire que les liens ont des flèches, comme sur l'exemple ci-dessus). Ils seront stockés dans un répertoire DATA sous forme compressée (et décompressés uniquement pendant leur utilisation). Ce répertoire contiendra un fichier CONTENT.txt listant les informations relatives à chaque réseau (nom complet ou autre nom du réseau, date d'ajout, provenance, nombre de nœuds et liens, etc). Les algorithmes seront stockés dans un répertoire ALGO qui contiendra également un fichier CONTENT.txt listant les informations relatives à chaque algorithme présent dans le système. Chaque algorithme est censé prendre en paramètre le chemin vers le fichier contenant le réseau (ou la description du réseau sur l'entrée standard si pas de paramètre) et produire en sortie leurs résultats sous forme de texte sur la sortie standard. Les résultats d'exécution des algorithmes seront sauvegardés dans des fichiers dans le répertoire RESULT.

Fonctionnalités à développer

Votre système devra intégrer les commandes suivantes (respecter les noms et paramètres indiqués, avec la possibilité d'ajouter d'autres paramètres si besoin) :

- `netadd <algo> | <réseau> ...` \mapsto ajout d'un algo (code C ou java à compiler ou bien exécutable) ou d'un réseau au système (fichier CONTENT.txt à mettre à jour).
- `netdel <algo> | <réseau>`
 \mapsto suppression d'un algo ou d'un réseau au système.
- `netinfo [<algo> | <réseau>]` \mapsto affichage des informations sur le système global (nombre de réseaux disponibles, nombre d'algorithmes, etc) si pas de paramètre ou bien sur un algo/réseau particulier si paramètre présent.
- `netrun <algo> | <lot d'algo> [<réseau> | <lot de réseaux>] ...` \mapsto exécution de l'algorithme ou du lot d'algorithmes sur le réseau ou le lot de réseaux voulu. Les lots d'algorithmes ou de réseaux seront définis dans des fichiers.
- `netscan` \mapsto vérification de l'intégrité du système (correspondance entre les fichiers CONTENT et les contenus des répertoires).

Les noms d'algo et de réseaux dans les commandes ci-dessus pourront être aussi bien les noms des fichiers correspondants que les autres noms contenus dans les fichiers CONTENT.

Pour tester votre système, vous récupérerez des réseaux des bases listées ci-dessus et vous développerez quelques algorithmes simples (dans les langages de votre choix), par exemple :

- test de la présence de boucle (un lien i-i qui relie un nœud avec lui-même)
- test de la présence d'arcs opposés (liens i-j et j-i)
- calcul des degrés (nombre de voisins) min moyen et max des nœuds
- comptage du nombre de triangles (liens i-j, j-k et k-i)
- etc

De plus, au moins une des options suivantes devra être implantée dans votre système :

Option 1 : récupération (semi)automatique de réseaux, à la demande, depuis le web ;

Option 2 : lancement des algorithmes sur serveur(s) distant(s) ;

Option 3 : génération de fichiers pdf de statistiques : statistiques par algorithme (tous les résultats de cet algorithme agrégés dans un fichier), statistique par réseau (agrégation des résultats de tous les algorithmes qui ont été exécutés sur ce réseau), ...

Modalités

- Le projet est à faire en binôme (demander explicitement l'accord du chargé de TP en cas d'autre configuration). La participation de chacun des membres du binôme devra apparaître sur les rapports et lors de la présentation.
- Le projet doit être réalisé en utilisant principalement les langages de scripts awk/shell et/ou en C.
- Un rapport préliminaire succinct (1 page, pdf uniquement) doit être envoyé par mél (à olivier.togni@u-bourgogne.fr) le 22 novembre au plus tard. Il contiendra, outre les noms du binôme, un premier aperçu de la structure générale de l'application et de la façon de réaliser les fonctionnalités demandées. La pertinence de ce rapport interviendra dans la note de rapport.
- Une démonstration du projet sera faite en salle machine (ou sur portable, pour ceux qui préfèrent) à partir du 16 décembre.
- Le rapport final doit être remis au plus tard au moment de la démo (sur papier ou par mél en pdf).