Université de Caen **Année 2008-2009**
UFR de Sciences **2ème session - 8 septembre 2009**

EI61T Approfondissement en algorithmique
durée 2h00

Les notes de cours et TD sont autorisées.
Les parties sont indépendantes, correspondent aux différentes parties du cours et doivent être rendues sur des copies séparées.

Chaque candidat doit, au début de l'épreuve, porter son nom dans le coin de la copie qu'il cachera par collage après avoir été pointé. Il devra en outre porter son numéro de place sur chacune des copies, intercalaires, ou pièces annexées.

1 Algorithmique répartie (à rédiger sur une copie séparée)

1.1 Introduction

On se place dans un réseau de processus ou sites communiquant par messages. Les lignes de communication entre sites sont supposées bi-directionnelles et fiables (tout message émis à une extrémité est reçu par l'autre extrémité en un temps fini). Les sites ne connaissent au départ que les lignes de communication les reliant à leurs voisins immédiats. Ces lignes sont, dans chaque site, repérées par un simple nombre entier allant de 1 au nombre de lignes de communications reliées à ce site (qu'on notera `nblignes`, cette constante pouvant bien entendu prendre des valeurs différentes selon les sites).

Un des sites, l'initiateur, joue un rôle particulier. Hormis ceci, les autres sites sont indifférenciés et n'ont pas d'identité.

1.2 Algorithme réparti à étudier

On considère l'algorithme réparti suivant :

```
L'initiateur lance l'algorithme en envoyant, sur chacune de ses lignes un message :
Pour tout i de 1 à nblignes faire
        parametre = "1-".str(i)  // le point est ici l'opérateur de concaténation,
         // str(i) est la représentation en caractères du nombre i
        Envoyer Message(parametre) via la ligne i
Nomsite = "1"

Tout site, sur réception de Message(p) via la ligne k :
Si aucun message relatif à cet algorithme n'a encore été reçu
        alors Nomsite = p
              Pere = k
              Pour tout i allant de 1 à nblignes, sauf k
                        parametre = p."-".str(i)
                        Envoyer Message(parametre) via la ligne i
        sinon // ne rien faire
```

Question 1.1 *En supposant que le réseau est connexe et fini, prouver que cet algorithme se termine en un temps fini et que tout site du réseau recevra un message en un temps fini.*

1.3 Exemple d'exécution

On considère le réseau de la figure 1 dans lequel l'initiateur est marqué par la lettre majuscule I.

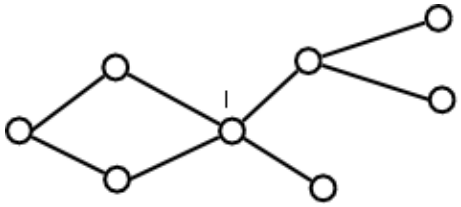

FIG. 1 – Exemple de réseau

Question 1.2 *Faire "tourner à la main" l'algorithme de la section 1.2 sur ce réseau. Pour cela, numéroter les lignes de communication de chacun des sites et décrire en détail au moins deux traitements de réception de message. Noter auprès des sites les valeurs, après exécution complète de l'algorithme, des variables* `Pere` *et* `Nomsite` *de chacun des sites du réseau.*

1.4 Cas général

On se place désormais dans le cas général, donc sur un réseau quelconque et non plus sur l'exemple de la section 1.3

Question 1.3 *L'algorithme de la section 1.2 ayant été exécuté sur un réseau,*
- *qu'obtient-on comme structure sur le graphe du réseau ?*
- *que représentent les valeurs obtenues par les variables* `Nomsite` *de chaque site relativement à cette structure ?*
- *les valeurs obtenues par les variables* `Nomsite` *sont elles deux-à-deux distinctes ?*

1.5 Utilisations

Question 1.4 *L'algorithme de la section 1.2 ayant été exécuté sur un réseau quelconque, décrivez comment :*
- *un site quelconque peut envoyer un message à l'initiateur*
- *un site quelconque peut connaître le nombre de liaisons que devra parcourir un tel message pour parvenir à l'initiateur (donc en fait sa distance à ce dernier).*

2 Algorithmique pour la Cryptologie (à rédiger sur une copie séparée)

Question 2.1 *Rappelez comment fonctionne le chiffrement RSA et précisez les algorithmes arithmétiques nécessaires à sa mise en œuvre, ainsi que leur complexité. Que pouvez-vous dire de sa sécurité du point de vue algorithmique ?*

Question 2.2

- *Calculez les deux derniers chiffres décimaux de* 3^{1000} *(indice : la réponse est* $3^{1000} \pmod{100}$*).*
- *Calculez* $11^2 \pmod{15}$*.*
- *Rappelez le théorème des restes chinois.*
- *Utilisez le théorème des restes chinois pour calculer différemment* $11^2 \pmod{15}$*.*

Question 2.3 *Soit* $N = pq$ *un entier produit de deux nombres premiers inconnus distincts* p *et* q*. Montrez que la connaissance de* N *et de* $\varphi(N) = (p-1)(q-1)$ *permet de retrouver* p *et* q*.*

3 Algorithmique des graphes (à rédiger sur une copie séparée)

Question 3.1
On donne les durées suivantes de trajets :

Nantes ↔ Paris-Montparnasse	*2 h*
Nantes ↔ Lyon	*7 h*
Paris-Montparnasse ↔ Paris-Lyon	*1 h (en autobus)*
Paris-Lyon ↔ Grenoble	*4 h 30*
Paris-Lyon ↔ Lyon	*3 h 30*
Marseille ↔ Lyon	*3 h*
Marseille ↔ Grenoble	*4 h 30*
Lyon ↔ Grenoble	*1 h*

Ces données peuvent être représentées par un graphe non orienté G *auquel on fera référence dans les exercices qui suivent.*

1. *Représenter le graphe* G*.*
2. *Indiquer la matrice d'adjacence associée au graphe* G*.*

Question 3.2
Appliquer l'algorithme de Dijkstra pour déterminer les plus courtes distances des sommets du graphe suivant au sommet A *:*