Université de Versailles – Saint-Quentin-en-Yvelines
Algorithmique Distribuée

TD 4 - Algorithmes Divers

Exercice 1. – *Compter dans un arbre* –

On considère un arbre *orienté* (un site distingue son père de ses fils) dans un système asynchrone. On souhaite calculer des informations relatives à cet arbre. Dans chacune des questions suivantes, on vous donne l'information que l'on souhaite calculer et vous devez écrire un algorithme distribué tel que, à la fin de l'exécution, la *racine* de l'arbre connait l'information désirée. On suppose que les initiateurs sont les feuilles de l'arbre.

1. Calculer le nombre de *feuilles* (*i.e.*, le nombre de nœuds sans fils) contenues dans l'arbre dans le cas où il est possible de mettre des valeurs dans les messages. Indiquer le mode de terminaison (implicite ou explicite, auquel cas vous indiquerez la (ou les) action(s) STOP_GLOBAL/STOP_LOCAL).
   Donner la complexité en messages et en temps de votre algorithme, dans le pire cas.

2. Calculer le nombre de *feuilles* contenues dans l'arbre dans le cas où il est interdit de mettre des valeurs dans les messages.

(i) Proposer dans un premier temps, un algorithme avec terminaison implicite.
Donner un exemple d'arbre dans lequel la complexité en message est $n-1$, puis un autre exemple dans lequel la complexité est $(n/2)^2$ (avec n, le nombre de nœuds dans l'arbre). En déduire la complexité en messages de votre algorithme, dans le pire cas. Quelle est sa complexité en temps dans le pire cas ?

(ii) Proposer dans un second temps, un algorithme avec terminaison explicite, en supposant des canaux FIFO. Puis, donner un exemple illustrant la nécessité d'avoir des canaux FIFO. Quelle est la complexité en messages et en temps de cet algorithme, dans le pire cas.

3. Calculer le nombre de noeuds *spéciaux* contenus dans l'arbre. Un noeud spécial est un noeud interne avec *exactement* un de ses fils qui est une feuille.

Exercice 2. – *Élection dans un graphe complet* –

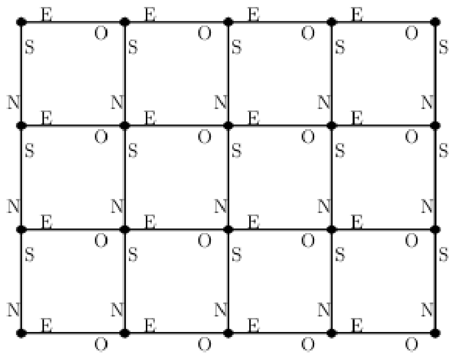
1. On rappelle qu'un graphe *complet* est un graphe dans lequel chaque site est relié à tous les autres sites. Sous l'hypothèse asynchrone, donner un algorithme d'élection dans un graphe complet et identifié. Voici les contraintes de l'algorithme :
   - On suppose que l'évènement *initialement* est généré sur un sous-ensemble non vide des sites du système.
   - Votre algorithme doit élire le site d'identifiant maximal ***parmi tous les sites initiateurs***.
   - À la fin de l'exécution, chaque site doit connaître l'identifiant du leader.
   - L'algorithme peut terminer sans qu'aucun site ne le sache.
2. Donner la complexité en messages et en temps de votre algorithme, dans le pire cas. Justifier brièvement.
3. Donner une version de l'algorithme dans laquelle le site finalement élu est capable de détecter sa propre terminaison (STOP_LOCAL).

Exercice 3. *– Problème de diffusion –*

1. Dans un graphe *quelconque*, il existe un site qui possède une information ($info_i$) très importante qui doit être envoyée à chacun des autres sites. Donner un algorithme pour la diffusion. Donner sa complexité en messages et en temps, dans le pire cas.

2. Si le graphe est une *grille orientée*, donner un algorithme de diffusion efficace en terme de messages envoyés. Une grille est orientée si sur chaque sommet, chaque arête incidente au sommet est étiquetée par la direction correspondante (Nord, Est, Ouest ou Sud), *cf.* la figure ci-dessous. Un site peut donc envoyer des messages dans les directions qu'il souhaite.

   Donner la complexité en messages et en temps de votre algorithme, dans le pire cas.



Exercice 4. *– Sur un tore –*

Un tore est un graphe de $n = t \times t$ sommets pour lequel chaque sommet peut se voir attribuer une identité définie par le couple (i, j) avec $0 \leq i, j < t$ de sorte que le sommet (i, j) et le sommet (i', j') sont voisins si et seulement si

- $(i = i')$ et $j = j' + 1$, on dira que (i', j') est en "bas" de (i, j)
- $(i = i')$ et $j = j' - 1$, on dira que(i', j') est en "haut" de (i, j)
- $(i = i' + 1)$ et $j = j'$, on dira que (i', j') est à "gauche" de (i, j)
- $(i = i' - 1)$ et $j = j'$, on dira que (i', j') est à "droite" de (i, j)

Attention, les additions et les soustractions sont modulo n (par exemple si $n = 3$, $1 + 2 = 0$, $0 - 1 = 2$)

1. Donner une illustration d'un tore avec $n = 3 \times 3 = 9$ sommets.
2. Un tore de $n = t \times t$ sommets est il un graphe régulier ? Si oui, donner le degré d'un tore.
3. Quel est le nombre d'arêtes d'un tore de $N = t \times t$ sommets.
4. Quel est le diamètre d'un tore de $N = t \times t$ sommets.

Dans la suite, on cherche à construire un algorithme efficace de diffusion sur un tore.

Indication: le tore est un graphe hamiltonien (*i.e.*, il est possible de construire un anneau sur le tore), la stratégie pour construire l'algorithme de parcours devra s'inspirer de ce résultat. Par exemple, à partir d'un site initiateur, l'idée est de rediriger le jeton uniquement vers les directions en "haut" et à "droite". Après k étapes, le jeton sera envoyé dans une direction si t est divisible par k et dans une autre sinon. L'algorithme termine lorsque le jeton revient sur l'initiateur.

5. Donner le pseudo-code de l'algorithme
6. Donner la complexité en message et la complexité en temps de l'algorithme, dans le pire cas.