

ENONCÉ DU TP 3

Sujet F

Printemps 2020

But du TP

Le but de ce TP est, avec le langage D, de travailler sur les graphes et :

- (1) d'implanter un algorithme d'apprentissage du graphe par les nœuds ;
- (2) d'implanter un algorithme de diffusion.

Dans tout le TP, vous supposerez que le graphe initial est une grille telle que générée dans l'exercice 4 du TP2. Les identifiants des nœuds ont été choisis aléatoirement dans $[0; n - 1]$ où n est le nombre de nœuds du graphe.

1 Exercice - Apprentissage du graphe sous-jacent

Question 1

Exécutez, à la main, l'algorithme permettant, à chaque nœud, de découvrir le graphe sous-jacent du système (algorithme vu en cours), sur une grille de 4×4 nœuds. Vous choisirez les identifiants des nœuds.

Question 2 Quand un nœud sait qu'il a terminé, c'est-à-dire qu'il a appris tout le graphe du système ?

Question 3 Décrire les structures de données que vous allez utiliser dans votre programme.

Question 4 Implantez cet algorithme.

Question 5 Mesurez expérimentalement le nombre de messages échangés avec votre pro-

gramme pour une grille de 4×4 , 10×10 et 50×50 et donnez les valeurs obtenues.

2 Exercice - Diffusion dans un graphe sans connaissance de la terminaison

Dans cet exercice, vous considérerez que l'opération de diffusion se fait avec l'algorithme d'inondation très simple où un nœud recevant le message à diffuser, pour la 1ère fois, le ré-émet vers tous ses voisins (sauf à celui dont il a reçu le message). C'est vous qui choisirez le nœud source de cette diffusion et non le système distribué.

Question 1 Implantez cet algorithme.

Question 2 Mesurez expérimentalement le nombre de messages échangés avec votre programme pour une grille de 4×4 , 10×10 et 50×50 et donnez les valeurs obtenues.

Vous avez la possibilité de retarder l'exécution d'un nœud en faisant dormir le thread associé pour un temps fixé. Ainsi, vous pouvez générer des exécutions très différentes de votre programme.

Aide : vous pouvez utiliser la fonction `sleep`

[https ://dlang.org/library/core/thread/osthread/thread.sleep.html](https://dlang.org/library/core/thread/osthread/thread.sleep.html).

Question 3 Incluez cette possibilité dans votre programme précédent. Est-ce que des exécutions différentes impliquant des endormissements de nœuds différents impliquent un nombre de messages échangés différents lors de l'opération de diffusion ?

Question 4 Si vous avez répondu positivement à la question précédente, mesurez expérimentalement le nombre de messages échangés sur 200 exécutions de votre programme sur une grille de 20×20 . Décrivez les scénarios choisis (choix de la source et des nœuds ralentis, valeurs choisies pour l'endormissement des nœuds) et donnez la distribution des valeurs obtenues sur le nombre de messages échangés.

3 Exercice - Diffusion dans un graphe avec connaissance de la terminaison

Vous voulez vous assurer, dans cet exercice, que la source de la diffusion sait que l'algorithme a terminé, c'est-à-dire que tous les nœuds ont bien reçu leur message.

Question 1 Modifiez l'algorithme de l'exercice précédent pour que la source puisse déterminer que l'algorithme a terminé. Décrivez votre algorithme.

Question 2 Implantez cet algorithme.

Question 3 Combien de messages supplémentaires sont ajoutés avec l'intégration de la connaissance de la terminaison à l'algorithme initial ?

Question 4 Supposons qu'initialement chaque nœud connaît le graphe sous-jacent grâce à l'algorithme de l'exercice 1. Proposez un algorithme, différent de celui proposé à la question 1, qui permet à la source de savoir que l'algorithme de diffusion a bien terminé.