PARIS DIDEROT

Examen
Durée : 3 heures (tout document autorisé)

M1 **Année 2018–2019**

Il sera tenu compte de la clarté de vos copies. Les questions avec des $\star$ seront particulièrement scrutées (plus de points). Les barèmes donnés sont indicatifs.

Exercice 1 : Algorithmes parallèles (5 points).

Une suite de Fibonacci est définie par $F_0 = 0$, $F_1 = 1$ et pour $n \geq 2$, $F_n = a.F_{n-1} + b.F_{n-2}$ avec $a > 0$ et $b > 0$.

1)- Ecrire un algorithme PRAM en mode EREW avec n processeurs capable de calculer F_n.

2)- Votre algorithme est-il efficace ?

3*)- Est-il optimal ? Dans le cas contraire, comment l'optimiser ?

Exercice 2 : Algorithme randomisé. (5 points)

On a un tableau T de longueur n contenant n entiers 2 à 2 distincts. Pour tout x dans T, on définit

$$\text{rang}(x) = 1 + \text{Card}\{y \in T,\, y < x\}\,.$$

4)- Etant donné un paramètre $\varepsilon > 0$, donner un algorithme déterministe qui renvoie un élément y de T dont le rang est proche de la valeur médiane :

$$\text{rang}(y) \in \left[\left(\frac{1}{2} - \varepsilon\right) n,\, \left(\frac{1}{2} + \varepsilon\right) n\right]\,.$$

5)- Quelle est la complexité de votre algorithme déterministe ?

6*)- Ecrire un algorithme Monte Carlo capable de répondre à la question 4 avec très forte probabilité.

7)- Quantifier votre algorithme probabiliste (complexité, taux de succès, ...). Quels sont les avantages de votre algorithme randomisé ?

Exercice 3 : Approximation d'approximation randomisée. (5 points)

Dans cet exercice, nous avons $x_1, \cdots, x_n$ n variables booléennes et une formule CNF est formée de m clauses $c_1, c_2, \cdots, c_m$ telle que chaque clause est la disjonction de 3 littéraux (x_i ou $\bar{x}_i$).
Le but est alors de trouver le nombre **maximum** de clauses contenant au moins un littéral à *vrai* et un littéral à *faux*. Ce problème est appelé : "Max Not-All-Equal-3-CNF".

8)- Ecrire un algorithme randomisé d'approximation pour le problème ci-dessus.

9*)- Quantifier le facteur d'approximation et analyser la complexité de votre algorithme.

10*)- Généraliser (définir "Max Not-All-Equal-k-CNF" et répondre aux mêmes questions).

Exercice 4 : Coupe minimum. (5 points)

11)- Définir le MinCut ou Coupe Minimum d'un graphe.

12)- Donner un algorithme capable de calculer le MinCut d'un arbre.

13*)- On considère l'algorithme suivant qui est une modification de l'algorithme de Karger.
Entrée : Graphe
Sortie : 2 sous-ensembles X et Y de sommets
A optimiser : le nombre d'arêtes entre X et Y doit être minimum.
Tant qu'il y a un nombre > 2 de sommets faire :
- choisir aléatoirement 2 sommets distincts et les fusionner en un seul sommet.
Les deux sommets restant à la sortie de la boucle "tant que" représentent alors les ensembles X et Y.
Montrer*** que le calcul du MinCut via cet algorithme nécessite un temps exponentiel (montrer que le taux de succès de cet algorithme est exponentiellement faible).

Exercice 5 : Election de leader dans un modèle distribué. (5 points)

Dans cet exercice, on considère un modèle de système distribué synchrone. On s'intéresse au réseau complet (tout nœud est connecté directement aux autres nœuds, i.e. le graphe sous-jacent est complet). A chaque instant t discret, un processeur (ou nœud) peut

- décider de déposer un message (de taille quelconque) sur le canal partagé
- écouter le canal partagé.

On suppose que le canal peut être consulté à tout moment t par tous les processeurs (qu'ils écoutent ou qu'ils envoient) et renverra les deux signaux suivants :

- SINGLE si et seulement si un seul processeur a envoyé un message à l'instant t
- NULL si 0 message a été déposé sur le canal ou si au moins 2 messages ont été déposés sur le canal (par au moins 2 processeurs).

Dans tout ce qui suit, on a les faits suivants.

- Le nombre total de processeurs n est connu de tous les processeurs.
- On admettra sans calcul que pour toute valeur de n, le produit

$$\prod_{i=1}^{4\log\log n}\left(1-\frac{n}{2^{2^k}}\left(1-\frac{1}{2^{2^k}}\right)^{n-1}\right)$$

  est strictement positif et strictement plus petit que 1 (et ce pour n petit ou grand).

14*)- k étant un entier positif, donner une interprétation de

$$\frac{n}{2^{2^k}}\left(1-\frac{1}{2^{2^k}}\right)^{n-1}$$

(Par exemple, en terme de probabilité.)

15*)- En déduire la signification de

$$\prod_{i=1}^{\textsc{Fin}}\left(1-\frac{n}{2^{2^i}}\left(1-\frac{1}{2^{2^i}}\right)^{n-1}\right)$$

("FIN" est un entier strictement positif.)

16*)-** En déduire un algorithme randomizé d'élection dans le modèle considéré et quantifier son temps de convergence (en moyenne, avec haute probabilité, ...).