COMPLEX - Session 2
Partie 2
2017-2018

Énoncé

Numéro d'anonymat :

Documents non autorisés. Seule une feuille A4 portant sur les cours et les TD est autorisée.
Téléphones portables éteints et rangés dans vos sacs.
Les exercices sont indépendants.
Une rédaction claire et concise sera appréciée. Toute affirmation devra être justifiée.

Exercice 1 : Classes de complexité probabilistes

1.a] Donner une définition formelle de la classe de complexité $\mathcal{ZPP}$

1.b] Donner une définition formelle de la classe de complexité $\mathcal{BPP}$

1.c] Donner une définition formelle de la classe de complexité $\mathcal{RP}$

Soient $T : \mathbb{N} \to \mathbb{N}$ une fonction et $L \subset \{0,1\}^*$ un langage. Nous dirons que L appartient à la classe $\widetilde{\mathcal{BPTIME}}(\mathcal{T})$, s'il existe une machine de Turing probabiliste M qui termine en temps espéré $T(|x|)$ pour tout mot $x \in \{0,1\}^*$ et telle que $\Pr[M(x) = 1] \geq 2/3$ si $x \in L$ et $\Pr[M(x) = 0] \geq 2/3$ si $x \notin L$ (où $\Pr[M(x) = b]$ pour $b \in \{0,1\}$ désigne la proportion des calculs de M qui retournent le résultat b sur l'entrée x). Nous notons $\widetilde{\mathcal{BPP}} = \bigcup_{k \geq 0} \widetilde{\mathcal{BPTIME}}(n \mapsto n^k)$.

1.d] Montrer que $\widetilde{\mathcal{BPP}} = \mathcal{BPP}$.

1.e] Montrer que si $\mathcal{NP} \subseteq \mathcal{BPP}$ alors $\mathcal{NP} = \mathcal{RP}$

Exercice 2 : Algorithme d'approximation probabiliste pour la 3-coloration maximale

En théorie des graphes, le problème de la 3-coloration est, étant donné un graphe non orienté $G = (V, E)$, de définir une fonction $c : V \to \{1, 2, 3\}$ telle que pour tout arête $\{u, v\} \in E$, $c(u) \neq c(v)$ (*i.e.* d'associer à chaque sommet une « couleur » dans l'ensemble $\{1, 2, 3\}$ de sorte que les sommets reliés par une arête soient de couleur differénte).

Le problème de la 3-coloration maximale pour un graphe non-orienté $G = (V, E)$, consiste à recherche une fonction $c : V \to \{1, 2, 3\}$ telle que l'ensemble

$$\{\{u, v\} \in E \quad | \quad c(u) \neq c(v)\}$$

est de cardinal maximal parmi toutes les colorations possibles. Un algorithme probabiliste simple pour ce problème consiste assigner à chaque sommet de G une couleur tirée uniformément aléatoirement dans $\{1, 2, 3\}$.

2.a] Donner le facteur d'approximation de cet algorithme probabiliste.

Exercice 3 : Arbre binaire ET-OU

Un *arbre binaire ET-OU* est un arbre binaire complet de profondeur n où chaque feuille est étiquetée soit 0, soit 1.

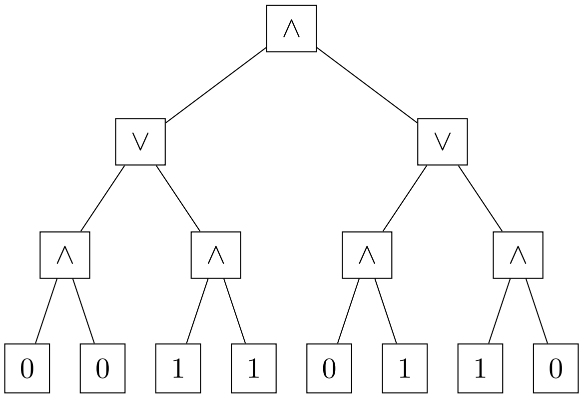
— La valeur d'une feuille (située à la profondeur n) est son étiquette.

— La valeur d'un nœud interne à la profondeur i est le « ou logique » ($\vee$) de la valeur de ces deux fils si i est de la même parité que n.

— La valeur d'un nœud interne à la profondeur i est le « et logique » ($\wedge$) de la valeur de ces deux fils si i est de la même parité que $n-1$.

Nous considérons le problème du calcul de la valeur de la racine d'un arbre binaire ET-OU étant donnée la suite des 2^n étiquettes de ses feuilles.

3.a] Calculer la valeur de l'arbre binaire ET-OU suivant :



3.b] Montrer que tout algorithme déterministe qui retourne la valeur d'un arbre binaire ET-OU doit examiner chaque feuille (et a une complexité en $\Omega(2^n)$).

Indication : Considérer le cas $n = 1$ puis utiliser un raisonnement par récurrence.

3.c] Proposer et analyser un algorithme probabibiliste (de type Las Vegas) qui retourne la valeur d'un arbre binaire ET-OU de profondeur n en temps espéré $O(c^n)$ pour une constante $c < 2$.
Indication : Considérer le cas $n = 1$ puis utiliser un raisonnement par récurrence.