

Algorithmique avancée – Examen Réparti 2
UPMC — Master d'Informatique —
Décembre 2010 – durée 2h

Les seuls documents autorisés sont les polys de cours, ainsi que la copie double personnelle.
Le barème, sur 23, est donné à titre indicatif.

Exercice 1 : Coût amorti (3 points)

Soit n un entier strictement positif et $b_{k-1} \dots b_1 b_0$ l'écriture de n en base 2. On note $\alpha(n)$ le plus petit indice i tel que $b_i = 1$. On suppose qu'un algorithme effectue une suite d'opérations, et que la n -ième opération coûte $2^{\alpha(n)}$.

Question 1. Montrer que, pour $n = 2^k - 1$ (avec $k \geq 1$), le coût total d'une suite de n opérations est égal à $k2^{k-1}$.

Question 2. En déduire le coût amorti d'une opération, lorsqu'on effectue une suite de n opérations.

Exercice 2 : Rotations (3 points)

Dans le processus de rééquilibrage d'un arbre (AVL ou bicolore), on utilise des opérations de *rotation*.

Question 1. Donner un exemple de rotation gauche (RG) et un exemple de rotation gauche-droite (RGD).

Question 2. Montrer que les opérations RG et RGD

- 1) conservent l'ordre infixe (symétrique)
- 2) diminuent de 1 la hauteur de l'arbre à la racine duquel est effectuée la rotation.

Exercice 3 : Algorithmes de Huffman (3 points)

La première question concerne l'algorithme de Huffman statique et la seconde l'algorithme de Huffman dynamique.

Question 1. On considère un texte dont les fréquences des lettres vérifient $p_1 \leq p_2 \leq \dots \leq p_n \leq 2p_1$. Montrer que l'arbre de Huffman statique correspondant à ces fréquences est équilibré (on pourra supposer que n est une puissance de 2).

Question 2. Décrire la suite des arbres de Huffman dynamiques obtenus pour la compression du texte **paradigite**, et donner le compressé.

Exercice 4 : Circuit polaire (4 points)

Soit S un ensemble de n points du plan, et $O \notin S$. Il s'agit ici de décrire le circuit polaire de S par rapport à O .

Question 1. Donner un algorithme, de complexité $O(n \log n)$, qui calcule le circuit polaire de S par rapport à O , en utilisant l'ordre polaire.

Question 2. Donner un algorithme, de complexité $O(n \log n)$, qui calcule le circuit polaire de S par rapport à O sans utiliser l'ordre polaire.

Question 3. Peut-on trouver un algorithme de calcul du circuit polaire d'un ensemble de n points, dont la complexité en nombre de comparaisons est $O(n)$? Argumentez votre réponse.

Exercice 5 : Code de Fibonacci (10 points)

Les nombres de Fibonacci sont définis par $F_0 = 0$, $F_1 = 1$, et pour $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$. La suite des nombres de Fibonacci, à partir de F_2 est 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

La valeur du n -ième nombre de Fibonacci F_n est égale à l'entier le plus proche de $\phi^n / \sqrt{5}$, avec $\phi = (1 + \sqrt{5})/2$.

Question 1. Tout nombre entier n strictement positif peut être représenté de manière unique par une somme de nombres de Fibonacci non consécutifs : $n = \sum_{i=0}^k F_{c_i}$, avec $c_0 \geq 2$ et $c_{i+1} > c_i + 1$. Par exemple l'unique représentation de 45 est $3 + 8 + 34$ (les sommes $3 + 8 + 13 + 21$ ou $1 + 2 + 8 + 34$ violent la condition "pas deux nombres de Fibonacci consécutifs"). Donner la représentation des nombres 32 et 105.

Décrire une méthode pour calculer la représentation d'un entier $n > 0$ (on pourra supposer que l'on dispose d'une table des nombres de Fibonacci).

Question 2. Le code de Fibonacci, inspiré de la représentation précédente, permet de représenter tout nombre entier strictement positif par une suite de bits qui se termine par 11 et ne contient jamais deux bits à 1 consécutifs ailleurs qu'à la fin. Par exemple les nombres 2, 3, 4, 5, 6, 7, 8 sont représentés par les codes 11, 011, 0011, 1011, 00011, 10011, 01011, 0000. Quels sont les codes des nombres 32, 45 et 105.

Expliquer comment est formé ce code.

Montrer qu'il y a F_r mots de code formés de $r + 1$ bits, pour $r \geq 1$.

Question 3. On représente maintenant une suite d'entiers strictement positifs, en concaténant leurs codes. Montrer que l'on peut décoder de manière unique une suite de bits correspondant à la concaténation de codes de Fibonacci : par exemple la suite 1011000111001101011 se décode en la suite d'entiers (4, 5, 6, 7).

Question 4. Décrire les algorithmes de codage et de décodage de Fibonacci, pour transformer une suite d'entiers en une suite de bits.

Question 5. Soit E une suite d'entiers, qui contient p_i fois l'entier n_i , et tel que la longueur du code de n_i est l_i bits. Montrer que la taille du codage de E est $\sum p_i l_i < 3 + 2H(E)$, où $H(E)$ est l'entropie de E .

Question 6. Que pensez-vous de cet algorithme pour la compression de texte ?

0 1 2 3 5 8 13 21 34
 0 1 1 1011
 10 11 100 101