L3 Informatique. Session 2 - Algorithmique

lundi 17 juin 2013. 08h00-10h00. T' 301.

La précision et la clarté de votre rédaction sont *fondamentales*. Documents interdits. Durée 2h00. Le barème indiqué est *approximatif*.

Exercice 1. [3pts] Le tableau $T = [9, 4, 7, 3, 5, 6, 8, 2, 1]$ est-il un tas ? Justifiez votre réponse. Si c'est le cas, échangez le premier terme avec le dernier terme puis appliquez l'algorithme descendre(T, n, k) du tri par tas pour rétablir la propriété APO du tableau. Si ce n'est pas le cas, indiquez en quels sommets la propriété APO n'est pas satisfaite et transformez ce tableau en TAS. Dans les deux cas dessinez l'arbre et son évolution.

Exercice 2. [1pt] Écrivez l'expression arithmétique suivante sous forme postfixée :

$$(12 - 3) \times (7 \times 2 + 5) - 3.$$

Exercice 3. [3pts] Dressez la table des longueurs des plus longues sous-séquences communes entre les mots `impossible` et `assimile`. Donnez une plus longue sous-séquence commune de ces deux mots.

Exercice 4. [6pts] On dispose d'un tableau $T[1, n]$ d'entiers triés dans l'ordre croissant. On se propose d'écrire un algorithme qui calcule l'indice p du tableau qui minimise la valeur

$$\left| \sum_{i=1}^{p} T[i] - \sum_{i=p+1}^{n} T[i] \right| \tag{1}$$

1. [1.5pts] Quelle est la valeur de p pour le tableau $T = [1, 1, 1, 1, 1, 1]$? Pour le tableau $T = [1, 2, 3, 4, 5, 6, 7]$? Pour le tableau $T = [1, 2, 4, 8, 16, 32, 64, 128, 256]$?
2. [3.5pts] Comment procéderiez-vous pour obtenir cet indice p ? Explicitez votre méthode en français. Écrivez ensuite votre algorithme barycentre(T) : entier dont le paramètre est un tableau de n valeur et qui renvoie l'entier p.
3. [1pt] En fonction de n calculez combien d'opérations ou de groupes d'opérations que votre algorithme effectue pour calculer la valeur de p.

Exercice 5. [7 points] On rappelle que pour a un élément d'un ensemble muni d'une loi multiplicative $\times$ et n un entier positif, on note

$$a^n = \underbrace{a \times a \times \cdots \times a}_{n \text{ fois}}$$

dont le calcul nécessite donc $n - 1$ multiplications.

4. [1pt] Expliquez comment l'on peut faire moins d'opérations en calculant les puissances successives $a^2 = a \times a$, $a^4 = a^2 \times a^2$, $a^8 = a^4 \times a^4$, etc. ?
5. [1pt] Combien de multiplications sont nécessaires dans ce cas ?
6. [2.5pts] Écrivez l'algorithme correspondant SM(a, n) : valeur qui effectue ce calcul.
7. [2.5pts] Implantez cet algorithme en langage C sous forme de fonction

   double SM(double a, ullong n)

   où ullong désigne le type unsigned long long. On rappelle les opérateurs bit-à-bit sur les entiers en C :

   (1) décalage de b bits à gauche `x << b`,

   (2) décalage de b bits à droite `x >> b`,

   (3) "ET" bit-à-bit `x & y`,

   (4) "OU" bit-à-bit `x | y`,

   (5) "NON" bit-à-bit, `~x`,

   (6) "OU EXCLUSIF" bit-à-bit `x ^ y`.