U.S.T.H.B - 30-01-2012
N. Bensaou - C. Ighilaza

Examen de rattrapage - durée 1h30 mn

Exercice 1.- 1. Montrer que: $n^{2^n} + 6 * 2^n = \Theta(n^{2^n})$

2. Montrez que si $f(n) = a_m n^m + a_{m-1} n^{m-1} + \cdots + a_1 n + a_0$ avec $a_0 > 0$ alors $f(n) = \Omega(n^m)$
3. Prouvez par induction sur n que:

$$\sum_{i=0}^{i=n} F_i = F_{n+2} - 1$$

où $F_i, i = 1, 2, \ldots, n, \ldots$ sont les nombres de Fibonacci [1]

Exercice 2.- 1. Soient A_1 et A_2 deux algorithmes qui résolvent le même problème, de complexité respective:

$$T_{A_1}(n) = n^2, \quad T_{A_2}(n) = 2^n/4$$

Pour quelles valeurs de n l'algorithme A_2 est préférable à A_1.

2. Soit f une fonction monotone décroissante et n le plus grand entier tel que $f(n) > 0$. On suppose que n existe.
   (a) Écrire un algorithme qui determine n. Donnez sa complexité.
   (b) Trouver un algorithme plus rapide que $\Theta(n)$.
3. Soit A un tableau de n composants entiers.
   (a) Écrire un algorithme qui determine le k-ième plus grand élément de A.
   (b) Donnez sa complexité.
   (c) Prouver sa validité.

Bon travail

[1] Les nombres de Fibonacci sont définis par la récurrence suivante:
$$F_n = \begin{cases} 1 & \text{si} \quad n = 0, 1 \\ F_{n-1} + F_{n-2} & \text{si} \quad n \geq 2 \end{cases}$$