Mathématiques pour l'informatique
Évaluation intermédiaire

Durée : 1 heure.

12 mars 2010

Exercice 1 : Ordres de grandeur
Soit les fonctions suivantes représentant des complexité d'algorithmes :

- $3n^3 + 2^{n-2}$.
- $4n^3 + 12$.
- $n^2 \log(5n^4)$.
- $\frac{1}{2}n^2 - 10n - 60$.
- $1/n$.

1. Trouvez les ordres de grandeur en notation $\mathcal{O}$ de chacune de ces fonctions.
2. Ordonnez ces ordres de grandeur du plus petit au plus grand.

Exercice 2 : Suite récurrente linéaire
On considère l'équation de récurrence suivante, pour une suite $(u_n)_{n\geq 0}$ de réels :

$$\forall n \in \mathbb{N},\ u_{n+2} = u_n + \frac{3}{2}u_{n+1}. \tag{1}$$

1. *Ensemble des solutions.* On note E l'ensemble des suites $(u_n)_{n\geq 0}$ qui vérifient l'équation de récurrence (1).
   Préciser la dimension de E et en donner une base.
2. *Étude matricielle.* Considérons la matrice à coefficients réels :

   $$A = \begin{pmatrix} 0 & 1 \\ 1 & 3/2 \end{pmatrix}$$

   (a) Montrer que la matrice A est diagonalisable sur $\mathbb{R}$. Trouver une matrice inversible $P \in M_2(\mathbb{R})$ telle que la matrice $P^{-1}AP$ est diagonale, et calculer A^n pour tout entier $n \geq 1$.

   (b) Soit $(u_n)_{n\geq 0}$ une suite à termes réels. Pour tout $n \in \mathbb{N}$, posons $X_n = \begin{pmatrix} u_n \\ u_{n+1} \end{pmatrix}$.
   Montrer que $(u_n)_{n\geq 0}$ vérifie l'équation de récurrence (1) si et seulement si $X_{n+1} = AX_n$ pour tout entier $n \geq 0$.

   (c) En déduire l'expression de u_n en fonction de n, u_0 et u_1.

Exercice 3 : Multiplication rapide : Karatsuba

Au milieu des années 50, Kolmogorov conjectura que la complexité minimum pour la multiplication était en $\mathcal{O}(d^2)$ (où d est la taille du plus grand entier qu'on souhaite multiplier). Cependant, en 1960, l'un de ses étudiants, appelé Karatsuba, découvrit une méthode en $\mathcal{O}(d^{\log_2(3)})$. L'objectif de cet exercice est d'expliciter cette méthode.

Soit donc $u = (u_{d-1} \dots u_0)_b$ et $v = (v_{d-1} \dots v_0)_b$ les écritures de u et v dans la base b que l'on suppose tous deux de taille $d = 2n$ pour simplifier.

Soit $R = b^n$ et écrivons

$$u = U_1 R + U_0, \quad v = V_1 R + V_0.$$

On a besoin de 4 multiplications et additions a priori pour réaliser ce produit.

1. Montrer que

$$uv = U_1V_1R^2 + ((U_0 + U_1)(V_0 + V_1) - U_1V_1 - U_0V_0)R + U_0V_0)$$

   En déduire que dans ce cas 3 multiplications et 6 additions suffisent pour calculer uv.

2. On utilise alors une stratégie récursive permettant de réduire la taille des facteurs U_i, V_i jusqu'à ce qu'ils soient suffisamment petits pour que la multiplication classique soit plus rapide (dans la pratique ce palier dépend essentiellement du processeur utilisé).
   Montrer que le temps de calcul t_d pour les données de tailles d est donné par la récurrence $t_d = 3t_{d/2} + 6n$.

3. Supposons $d = 2^m$. La récurrence devient alors

$$t_0 = 1, \quad t_m = 3t_{m-1} + 6.2^{m+1}$$

   En déduire, à l'aide des fonctions génératrices, que le temps de calcul est alors

$$t_d = 11n^{1,58496} - 12n.$$

   Indication : On a $\log_2(3) = \frac{\ln 3}{\ln 2} \approx 1,58496$.