ÉNS de Lyon — Option C
2018-2019 — Agrégation

TP : ALGORITHMES ÉLÉMENTAIRES

Exercice 1 —Algorithme de Karatsuba Soient $n \in \mathbb{N}$, $m = \lceil \frac{n}{2} \rceil$, $a, b \in [\![0, 2^n - 1]\!]$. Soient α le quotient de la division euclidienne de a par 2^m et $\tilde{\alpha}$ le reste de la division euclidienne de a par 2^m. On définit de même β et $\tilde{\beta}$. On rappelle que $ab = 2^{2m}\alpha\beta + 2^m\big(\alpha\beta + \tilde{\alpha}\tilde{\beta} - (\alpha - \tilde{\alpha})(\beta - \tilde{\beta})\big) + \tilde{\alpha}\tilde{\beta}$.

1. Programmer un algorithme « kar » de multiplication de a par b utilisant cette méthode.

2. Programmer un algorithme « usu » de multiplication de a par b utilisant la méthode usuelle.

3. Programmer un algorithme prenant en entrée un entier n et qui renvoie le temps de calcul de deux nombres aléatoires a et b compris entre 2^{n-1} et 2^n par les algorithmes « kar » et « usu ».

4. Que se passe-t-il lorsque l'on produit un nombre supérieur à 2^{63} avec «randint». Programmer un algorithme qui prend en entrée un entier n et qui renvoie deux nombres aléatoires a et b compris entre 2^{n-1} et 2^n.

Exercice 2 —Algorithme d'Euclide

Pour tous entiers $n, m \in \mathbb{N}^*$, on note $\tau(m, n)$ le nombre de divisions nécessaires dans l'algorithme d'Euclide classique pour calculer le pgcd de m et n, en commençant par la division de m par n. On a donc par exemple $\tau(1, 2) = 2$ et $\tau(2, 1) = 1$.

1. Ecrire un algorithme permettant de calculer $\tau(m, n)$.

2. Désignons par $(F_n)_{n\geq 0}$ la suite de Fibonacci, avec $F_0 = 0$ et $F_1 = 1$.

(a) A l'aide de l'algorithme précédent, calculer les 50 premières valeurs de $\tau(F_n + 2, F_{n+1})$.

(b) Démontrer que pour tout $n \geq 1$, on a $\tau(F_{n+2}, F_{n+1}) = n$.

3. Écrire un algorithme permettant de vérifier que $\max\{\tau(m, n), 0 < m, n < N\}$ est atteint par le couple (F_r, F_{r+1}), où r désigne le plus grand entier tel que l'on ait $F_{r+1} < N$.

4. Tester cet algorithme sur quelques exemples, puis prouver que la propriété énoncée dans la question précédente est effectivement vraie.

5. Posons $\alpha = \frac{1+\sqrt{5}}{2}$ et $\beta = \frac{1-\sqrt{5}}{2}$.

(a) Exprimer F_n en fonction de α et de β.

(b) Vérifier que l'on a $|\beta^n| \leq \frac{\sqrt{5}}{2}$.

(c) En déduire que F_n est l'entier le plus proche de α^n.

(d) Démontrer que pour tous entiers m, n vérifiant $0 < m, n < N$, on a $\tau(m,n) \leq \frac{\ln(\sqrt{5}N)}{\ln(\alpha)}$.

6. On désigne par φ la fonction indicatrice d'Euler. Pour tout entier $n \geq 1$, notons I_n l'ensemble des entiers $k \in [\![1, n-1]\!]$ premiers avec n et posons alors

$$T_n = \frac{1}{n}\sum_{k=1}^{n-1}\tau(k,n) \text{ et } \tau(n) = \frac{1}{\varphi(n)}\sum_{k\in I_n}\tau(k,n).$$

(a) Représenter les graphes des suites T et τ sur le même repère. Que constatez-vous ?

(b) Vérifier graphiquement qu'il existe un entier n_0 pour lequel $0,843\ln n + 1,47 + n_0$ est une bonne approximation de $\tau(n)$. Quel est l'ordre de grandeur de n_0 ?

Exercice 3 —[Réciproque du théorème chinois]

1. Programmer un algorithme qui prend en entrée deux listes $M = [m_0, \ldots, m_k]$ et $[x_0, \ldots, x_k]$, où les m_i sont des entiers deux à deux premiers entre eux et les x_i sont des éléments de $\mathbb{Z}/m_i\mathbb{Z}$, et qui renvoie un élément x tel que $(x[m_0], \ldots, x[m_k]) = (x_0, \ldots, x_k)$, par la méthode de Knuth.

2. Programmer un algorithme qui donne les mêmes résultats, mais en calculant $m'_i = \prod_{j\neq i} m_i$ puis $e_i = b_i m'_i$, où $b_i m'_i \equiv 1[m_i]$.