Maitrise et IUP d'informatique 1999-2000
Université de Marne la Vallée
Calculabilité. Devoir 2

Le devoir est à rendre pour le 15 Janvier 2000. Le devoir peut se faire par groupe d'au plus deux étudiants. La programmation peut s'effectuer dans le langage de programmation de votre choix, C, C++, Java, Maple. La programmation nécessite la manipulation d'entiers longs. On ne demande pas de reprogrammer les fonctions de base sur les entiers longs mais d'utiliser les bibliothèques ou API.

Exercice 1.— Implémenter l'algorithme probabiliste que constitue le test de Miller-Rabin qui permet de tester si un entier n est premier. Ce test est donné dans le transparent 136 du cours accessible à
`http://www-igm.univ-mlv.fr/~beal/Enseignement/Calculabilite/`.
Des variantes de cet algorithme sont données dans la référence [2] page 425-426. On demande de faire des tests sur des entiers longs. On pourra vérifier les résultats avec la fonction `isprime()` de Maple.

Exercice 2.— On peut montrer que si n est un nombre composite, il existe au moins $\frac{3}{4}$ de vrais temoins parmi les temoins possibles. Vérifier expérimentalement ce résultat sur des nombres composés qui sont cette fois-ci petits. Ce taux varie-t-il beaucoup d'un nombre à l'autre ? Calculer le taux de temoins pour le nombre composite $n = 200043671$.

Exercice 3.— Alice et Bob conservent chacun une copie d'une même base de données. Les deux copies peuvent s'interprêter comme deux longues suites de n-bits qui forment deux entiers $a = \sum_{i=1}^{n} a_i 2^{i-1}$ et $b = \sum_{i=1}^{n} b_i 2^{i-1}$. On a $a, b \leq 2^n$. Alice et Bob vérifient périodiquement que l'une des bases n'a pas été altérée par Eve sans vérifier les n bits. Pour cela, on définit $F_p(x) = x \bmod p$ pour un nombre premier p. Alice calcule $F_p(a)$, l'envoie à Bob qui calcule $F_p(b)$ et compare avec l'empreinte $F_p(a)$ d'Alice.

Le nombre p choisi est petit par rapport à 2^n. Alice ou Bob tire au hasard un nombre premier p inférieur à τ. La taille des messages échangés par Alice et Bob est ainsi de l'ordre de $\log(\tau)$ et non plus de n. On peut montrer que le nombre de nombres premiers distincts diviseurs d'un entier inférieur à 2^n est au plus n. On en déduit que la probabilité d'erreur vérifie :

$$\mathbf{Pr}[F_p(a) \neq F_p(b) \mid a = b] \leq \frac{n}{\pi(\tau)},$$

où $\pi(\tau)$ est le nombre de nombres premiers inférieurs à τ, de l'ordre de $\frac{\tau}{\log(\tau)}$. Quel choix pour τ et donc pour le tirage aléatoire des nombres premiers peut être interessant ? Le nombre τ doit être assez grand pour que la probabilité d'erreur soit faible et assez petit pour que la taille des echanges entre Alice et Bob soit petit devant n (de l'ordre de $\log(n)$). Programmer ce protocole de vérification de la non altération d'une base de données.

Références

[1] *Codes and Cryptography*, D. Welsh, Clarendon Press, Oxford,1988.

[2] *Randomized Algorithms*, R. Motwani and P. Raghavan, Cambridge University Press, 1995a (Disponible à la bibliothèque info-recherche. Venir me voir si vous n'y avez pas acces).