Exponentiation rapide

Les algorithmes efficaces pour tester si un nombre est premier, ou pour le décomposer en facteurs premiers, ainsi que d'autres algorithmes fondamentaux en cryptographie, utilisent le calcul de puissances a^e (**mod** n) pour de grandes valeurs de l'exposant e. Il existe en effet un algorithme très simple pour effectuer rapidement un tel calcul : sa complexité (comptée en nombre de multiplications à effectuer) est une fonction linéaire du nombre de chiffres de l'exposant. Cet algorithme est valide pour calculer n'importe quelle puissance, pas forcément en arithmétique modulaire ; mais c'est dans cette situation qu'il est le plus souvent utilisé ; on l'appelle *élever au carré et multiplier* :

```
long puissance (long a, long e, long n) {
  long p;

  for (p = 1; e > 0; e = e / 2) {
    if (e % 2 != 0)
      p = (p * a) % n;
    a = (a * a) % n;
  }
  return p;
}
```

Cette fonction a priori mystérieuse repose sur l'écriture de l'exposant e en numération binaire. Si par exemple e s'écrit 1011001, on a :

$$e = 1 + 8 + 16 + 64 = 89 \text{ et } a^e = a * a^8 * a^{16} * a^{64}$$

La suite $a, a^2, a^4, a^8, a^{16}, a^{32}, a^{64}$ s'obtient par une suite d'élévations au carré. Pour calculer $p = a^e$ il suffit donc, pour chacun des chiffres binaires de l'exposant e (lus de droite à gauche), d'effectuer les opérations suivantes :

1. multiplier p par a si le chiffre binaire vaut 1 ;
2. élever a au carré.

Commentaires de programmation

- Le calcul des chiffres binaires de l'exposant e utilise l'algorithme classique exposé dans la leçon sur la numération : ces chiffres sont les restes des divisions successives par la base (ici 2).
- Cette fonction donne des résultats incorrects si l'un des produits déborde avant la réduction modulo n ; telle quelle, elle ne peut donc être utilisée que pour des valeurs de n inférieures à 2^{16}. Pour des valeurs supérieures, il faudrait employer des fonctions de calcul en multiprécision, au lieu des opérations effectuées directement par le processeur.

Complexité

Le nombre total de multiplications est égal au nombre de chiffres de l'exposant e en numération binaire (pour les élévations au carré), plus le nombre de chiffres égaux à 1 ; soit $7 + 4 = 11$ multiplications pour calculer a^{89}. Cette complexité est donc *linéaire* en fonction du *nombre de chiffres* de l'exposant, autrement dit proportionnelle au *logarithme* de l'exposant.

Voir aussi

Nombres premiers (test de Fermat), cryptographie.