in. E. FOURREY, Récréations Arithmétiques
réimpression de l'édition de 1899, Vuibert, 1994, VII à XIX.

DES APPLICATIONS DE L'ARITHMÉTIQUE À L'INFORMATIQUE

PAR JEAN-LOUIS NICOLAS

C'est une excellente initiative de rééditer les récréations arithmétiques d'Émile Fourrey. L'arithmétique n'est plus guère enseignée de nos jours, mais quantité de revues scientifiques de tous niveaux ont un grand succès en parlant de problèmes de nombres. Différentes compétitions ou rallyes regroupent sur le thème mathématique de nombreux lycéens, et même des collégiens et l'arithmétique joue, dans ces joutes plaisantes, un rôle non négligeable. Enfin, les ordinateurs influent fortement sur cet attrait, car l'arithmétique est facile à programmer et, en revanche, l'ordinateur exécute les opérations ennuyeuses en laissant à l'exécutant le plaisir des résultats.

Émile Fourrey aurait certainement beaucoup apprécié le problème dit « 3 x + 1 ». Vous prenez un nombre ; s'il est pair, vous le divisez par deux ; s'il est impair, vous le multipliez par 3 et vous ajoutez 1. Et puis, on recommence. On constate que l'on tombe sur 1 au bout d'un temps plus ou moins court.

Exemple :

13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

La suite qui démarre à 27 est beaucoup plus longue. Chacun peut faire à la main, ou avec une machine, quelques essais. Mais voilà, on ne sait pas démontrer que l'on tombe toujours sur 1. On l'a vérifié dans de nombreux cas, mais on n'a pas de preuve mathématique.

L'informatique a remis à la mode certaines parties de l'arithmétique, par exemple, l'écriture des nombres avec les chiffres. On a réfléchi à différentes possibilités de représenter des nombres dans l'ordinateur, mais la bonne veille méthode d'écrire les nombres avec leurs chiffres dans une base (base 10 pour les hommes, base 2 pour les ordinateurs) est toujours fidèle au poste. Les nombreux problèmes posés par E. Fourrey

sur « les nombres abstraits », dans la première partie de son ouvrage, inspireront peut-être des informaticiens pour améliorer un de leurs logiciels. Le problème 73, qui étudie une famille optimale de poids, est un bon problème d'algorithmique.

Je veux traiter ici de deux questions : le calcul des puissances, et la méthode de factorisation « ro » de Pollard. Ces deux sujets sont à la fois simples et attrayants et, de plus, ils sont d'une très grande utilité : le calcul des puissances a comme conséquence les tests de primalité modernes (savoir reconnaître qu'un nombre est premier) et la méthode de cryptographie (transmission secrète des messages) RSA qui seront brièvement évoqués ci-dessous. Quant à la méthode « ro », ce qu'elle a de surprenant, c'est sa simplicité et le fait qu'elle n'est pas issue de l'arithmétique à proprement parler.

L'algorithme des puissances

Si l'on veut calculer 2^{16}, on peut multiplier 2 par 2, puis le résultat par 2, puis, etc., et au bout de 15 multiplications, obtenir 2^{16}. Mais on peut aussi calculer $2^2 = 4$, puis $4^2 = 16$, puis $16^2 = 256$, puis $256^2 = 65\,536$ et obtenir le résultat en seulement 4 multiplications, en utilisant l'égalité

$$\left(\left(\left(2^2\right)^2\right)^2\right)^2 = 2^{16}$$

Si l'on veut calculer $2^{1\,024}$, sachant que $1\,024 = 2^{10}$, au lieu de 1 023 multiplications, par la première méthode, il suffira de 10 opérations par la seconde, ce qui est un gain appréciable.

Oui, mais si l'exposant n'est pas lui-même une puissance de 2 comment faire ? On utilise alors l'écriture de l'exposant en base 2 :

Pour calculer a^b.

1. Écrire b en base 2.
2. Supprimer le « 1 » à gauche.
3. Remplacer « 1 » par CM et « 0 » par C.
4. On part de a, et on effectue les opérations à partir de la gauche. C est l'élévation au carré, M est la multiplication par a.

Exemple : calcul de 2^{23}.

1. 23 s'écrit en base 2 : 10111
2. Supprimer le « 1 » à gauche 0111
3. C CM CM CM
4. 2 C 4 C 16 M 32 C 1024 M 2048 C 4194304 M 8388608

La justification peut se faire par récurrence sur le nombre de chiffres de l'exposant b en base 2. Pour les lecteurs un peu plus savants c'est en fait une variante de l'algorithme de Hörner pour évaluer un polynôme.

Pour évaluer le polynôme

$$ax^3 + bx^2 + cx + d,$$

le schéma de Hörner l'écrit :

$$((ax + b)\,x + c)\,x + d.$$

De façon semblable, l'écriture de 23 en base 2 permet d'écrire :

$$23 = 2^4 + 0 \cdot 2^3 + 2^2 + 2 + 1 = (((2 + 0)\,2 + 1)\,2 + 1)\,2 + 1$$

et, pour calculer a^{23}, on calcule a^{2+0}, puis $a^{2\cdot 2+1} = a^5$, puis $a^{5\cdot 2+1} = a^{11}$, et enfin $a^{11\cdot 2+1} = a^{23}$.

Dans le problème 158, qui est plus difficile que la moyenne, Émile Fourrey étudie le chiffre terminal d'une puissance. Si l'on regarde les puissances successives de 1994, on voit qu'elles se terminent alternativement par 4 et par 6. Si l'on regarde les deux derniers chiffres des puissances de 1994, on voit que 1994^2 et 1994^{12} se terminent tous deux par 36, et qu'il y a ainsi une période de longueur 10.

Et si l'on veut connaître les quatre derniers chiffres de 1994^{13} ? On peut utiliser l'algorithme des puissances, mais en ne conservant que les quatre derniers chiffres des nombres utilisés.

Calculer les quatre derniers chiffres de 1994^{13}.

Le calcul peut se faire avec une calculette « quatre opérations » :

1. Écrire 13 en base 2 : 1101
2. Supprimer le « 1 » à gauche : 101
3. Remplacer 1 par CM, 0 par C : CMCCM

(C est l'élévation au carré, M est la multiplication par 1 994).

4. Rentrer 1 994.

Effectuer le premier C : $1\,994^2 = 3\,976\,036.$
Garder les quatre derniers chiffres : 6 036.
Effectuer le premier M : $6\,036 \times 1\,994 = 12\,035\,784.$
Garder les quatre derniers chiffres : 5 784.
Effectuer C : $5\,784^2 = 33\,454\,656.$
Garder les quatre derniers chiffres : 4 656.
Effectuer C : $4\,656^2 = 21\,678\,336.$
Garder les quatre derniers chiffres : 8 336.
Effectuer le dernier M : $8\,336 \times 1\,994 = 16\,621\,984.$

Garder les quatre derniers chiffres : 1 984.

5. Résultat : les quatre derniers chiffres de $1\,994^{13}$ sont 1 984.

En janvier 1994, D. Slowinski vient de battre le record du monde du plus grand nombre premier connu. Il s'agit du nombre $N = 2^{859\,433} - 1$. Le logarithme décimal de ce nombre vaut :

$$\log_{10}(N) = 258\,715{,}1\,123.$$

C'est donc un nombre de 258 716 chiffres décimaux.

Comme $10^{0{,}1\,123} = 1{,}29\ldots$, on voit que les trois premiers chiffres de N sont 129. Par la méthode ci-dessus, on peut calculer ses quatre derniers chiffres décimaux.

Le test de Fermat

Soit p un nombre premier, et a un nombre plus petit que p. Le petit théorème de Fermat affirme que :

$$a^{p-1} \equiv 1 \bmod p$$

autrement dit, que p divise $a^{p-1} - 1$.

Soit n un nombre impair. On dit que n passe le test de Fermat en base a si l'on a :

$$a^{n-1} \equiv 1 \bmod n$$

autrement dit, si n divise $a^{n-1} - 1$.

D'après l'algorithme des puissances décrit plus haut, ce test est très rapide à exécuter : on écrit $n - 1$ en base 2, on supprime le 1 à gauche, on remplace 0 par C, 1 par CM, mais cette fois-ci, C est l'élévation au carré modulo n, c'est-à-dire que l'on élève au carré, puis on prend le reste de la division par n. De même, M est la multiplication par a modulo n : on multiplie par a, puis on prend le reste de la division par n. Pour un nombre n de 100 chiffres décimaux, le logarithme décimal de n est, à une unité près, égal à 100, le logarithme en base 2 de n est 100/0,30 103... = 332,193, et le nombre de chiffres de n en base 2 est certainement inférieur à 350. Le nombre d'opérations C ou M à effectuer est donc inférieur à 700. On peut donc effectuer le test de Fermat pour un nombre n de 100 chiffres au prix de moins de 700 opérations, chacune constituée d'une multiplication de deux nombres de 100 chiffres au plus et suivie d'une division par n. Ceci s'effectue très vite en ordinateurs.

D'après le petit théorème de Fermat, si un nombre n plus grand que a ne passe pas le test de Fermat en base a, c'est-à-dire si

$$a^{n-1} \not\equiv 1 \bmod n,$$

alors, certainement, n est composé, ce qui veut dire non premier. Il est donc facile de montrer qu'un nombre n est composé, mais cette méthode ne fournit pas de diviseur de n.

Par contre, il existe des nombres composés qui passent le test de Fermat. Lorsque $a = 2$, le plus petit est $n = 341 = 11 \cdot 31$. Ce nombre est dit pseudo-premier en base 2 ($pp - 2$). C. Pomerance, J. Selfridge et S. Wagstaff ont calculé que jusqu'à 25 milliards, il y avait exactement 21 853 nombres $pp - 2$, tandis qu'il y a un peu plus d'un milliard de nombres premiers. On voit donc que si un nombre au hasard inférieur à 25 milliards passe le test de Fermat, la probabilité qu'il soit composé est inférieure à $22\,000 \cdot 10^{-9}$, ce qui est très petit.

L'existence de ces nombre pseudo-premiers a considérablement compliqué la recherche de tests de primalité. François Morain a écrit un programme basé sur la méthode d'Atkin utilisant les courbes elliptiques (méthode qui nécessite des développements mathématiques de haut niveau) et qui permet de prouver rigoureusement qu'un nombre jusqu'à 1 500 chiffres décimaux est premier.

Le protocole de cryptographie RSA

Cette méthode d'échange de messages secrets tire son nom de ses trois inventeurs Rivest, Shamir, Adleman, et date de 1978. Le chef de réseau Alice exécute les opérations suivantes pour mettre en place le protocole :

1. Construire deux nombres premiers p et q de 100 chiffres environ.
2. Calculer $n = pq$.
3. Calculer $\varphi = (p - 1)(q - 1)$.
4. Choisir e au hasard, premier avec φ, et plus petit que φ.
5. Calculer d, tel que $ed \equiv 1 \bmod \varphi$ (ceci est possible grâce à l'algorithme d'Euclide étendu).
6. Publier n et e. Garder secret p, q, φ et d.

Pour envoyer un message à Alice, Bob le met sous la forme d'un (ou plusieurs) nombre(s) M plus petit(s) que n. Il calcule ensuite par l'algorithme des puissances :

$$C \equiv M^e \bmod n$$

ce qui est facile puisque e et n sont publics. Pour lire la lettre de Bob, Alice calcule $C^d \bmod n$, grâce au nombre secret de déchiffrement d qu'elle connaît. On peut démontrer que

$$M = C^d \bmod n.$$

Pour le moment, on ne sait pas factoriser de grands nombres. La taille maximum des nombres que l'on sait systématiquement décomposer en facteurs premiers ne dépasse guère une centaine de chiffres. Par conséquent, un ennemi qui espionne Alice et Bob ne saura pas trouver p et q à partir de n (qui a environ 200 chiffres).

Mais, si l'on savait factoriser des nombres d'une certaine taille en un temps comparable à celui de construire des nombres premiers de même taille, alors ce protocole RSA ne serait plus d'aucune utilité.

La découverte de cette procédure RSA a vivement intensifié les recherches en arithmétique, notamment sur les nombres premiers et les algorithmes de factorisation. Nous présentons ci-dessous l'un de ces algorithmes, mais il faut d'abord évoquer le paradoxe des anniversaires qui est à la base du fonctionnement de la méthode « ro » de Pollard.

Le paradoxe des anniversaires

On supposera dans ce qui suit qu'il n'y a pas d'années bissextiles ou, plus exactement, qu'on exclut du jeu les personnes nées un 29 février. On supposera également que les naissances se répartissent équitablement entre les 365 jours de l'année.

Le principe des tiroirs affirme que, si une commode comprend k tiroirs, et contient $k + 1$ chaussettes, un des tiroirs contient au moins 2 chaussettes. C'est ce principe qui est utilisé dans le problème 213, p. 165, du livre d'Émile Fourrey, sur le nombre de cheveux.

On réunit n personnes. Si $n \geq 366$, d'après le principe des tiroirs, il y a sûrement deux personnes qui ont leur anniversaire le même jour. Mais si $n \leq 365$, on peut toujours choisir n personnes avec des anniversaires différents.

Le paradoxe des anniversaires est que si l'on réunit au hasard 23 personnes seulement, la probabilité que deux personnes aient même jour d'anniversaire est plus grande que celle de l'événement contraire, c'est-à-dire que les 23 personnes aient des jours anniversaires distincts.

Justifions l'assertion ci-dessus. Le nombre de cas possibles est 365^{23}. Comptons le nombre de cas où les anniversaires sont tous différents, et faisons rentrer les 23 personnes une à une. Pour la première, son jour anniversaire importe peu, il y a 365 cas favorables.

Pour la deuxième, le seul impératif est que son jour anniversaire soit différent de celui de la première personne, soit 364 cas.

Pour la troisième, 363 cas favorables, etc.

Soit, au total,

$$N = 365 \times 364 \times \ldots \times (365 - 23 + 1) \text{ cas.}$$

On calcule $p = N/(365)^{23} = 0{,}49\,270 < 1/2$.

Il y a donc une probabilité $1 - p > 1/2$ que deux personnes au moins aient leur anniversaire le même jour.

Posons $m = 365$, et supposons qu'il y ait n personnes, avec $n < m$.

La probabilité ci-dessus s'écrit :

$$p = p(m, n) = \frac{m(m-1)\ldots(m-n+1)}{m^n} = \prod_{k=1}^{n-1}\left(1 - \frac{k}{m}\right).$$

En utilisant l'inégalité $\log(1 + x) \leq x$, valable pour tout $x > -1$, on obtient :

$$\log p\,(m, n) = \sum_{k=1}^{n-1} \log\left(1 - \frac{k}{m}\right) \leq -\frac{1}{m}\left(\sum_{k=1}^{n-1} k\right) = -\frac{n(n-1)}{2m}$$

et encore :

$$p(m, n) \leq \exp\left(-\frac{n(n-1)}{2m}\right).$$

En fait, lorsque n n'est pas trop grand, l'inégalité ci-dessus est presque une égalité : pour $m = 365$ et $n = 23$, on a :

$$\exp\left(-\frac{n(n-1)}{2m}\right) = 0{,}49\,999\,825, \text{ et } p(m, n) = 0{,}49\,270.$$

Lorsque l'on réunit 60 personnes, l'inégalité ci-dessus donne :

$$p(365{,}60) \leq \exp\left(-\frac{60 \cdot 59}{2 \cdot 365}\right) = 0{,}00783\ldots$$

et il y a moins d'une chance sur 100 que les 60 personnes aient leur anniversaire tous différents.

Le paradoxe des anniversaires a trouvé de nombreuses applications en mathématiques et en informatique, en particulier la méthode « ro » de Pollard.

Le théorème « de la poële à frire »

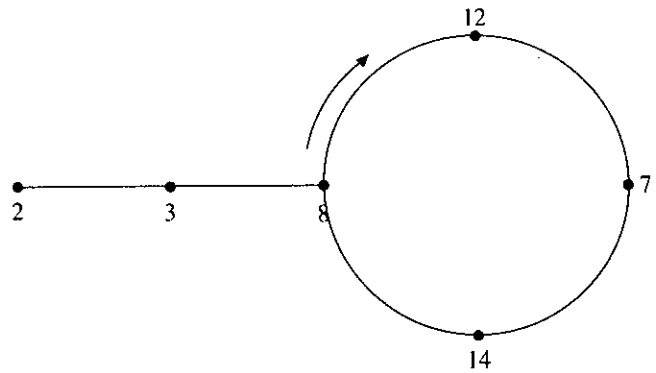
Soit E un ensemble à m éléments, par exemple $E = \{0, 1, \ldots, m-1\}$ et f une application de E dans E. On part de x_0 fixé dans E et l'on calcule $x_1 = f(x_0)$, $x_2 = f(x_1)$, etc. Nous allons montrer que la suite x_n ainsi obtenue est périodique après quelques termes exceptionnels.

Exemple : Prenons $m = 17$, et $f(x) = x^2 - 1 \bmod 17$, c'est-à-dire le reste de la division de $x^2 - 1$ par 17. On part de $x_0 = 2$.

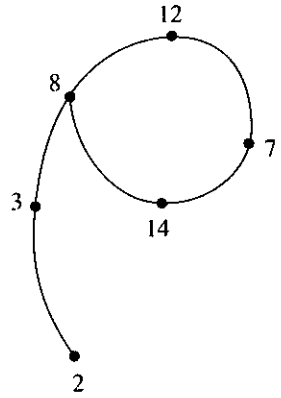
On obtient successivement : $x_1 = 3$
$x_2 = 8$
$x_3 = 12$
$x_4 = 7$
$x_5 = 14$
$x_6 = 8$

et comme $x_6 = x_2$, on aura $x_7 = x_3$, ... et l'on voit ainsi que la suite « cycle ».

Ceci peut se dessiner :



et l'on voit apparaître la poële à frire avec la queue (2, 3) et le cycle (8, 12, 7, 14). Les Anglais, moins bassement intéressés par leur estomac, préfèrent dessiner la lettre grecque ro :



Le théorème de la poële à frire va montrer que la situation présentée dans l'exemple ci-dessus est générale. La philosophie que l'on peut en déduire est que « tout processus fini est ultimement périodique », c'est-à-dire que, lorsque le nombre de situations possibles est fini, on retombe toujours, au bout d'un certain temps, sur une situation déjà rencontrée.

Théorème : *Soit E un ensemble fini à m éléments, f une application de E dans E, x_0 un élément de E. On construit la suite $x_1 = f(x_0)$, $x_2 = f(x_1)$, etc. Il existe deux constantes $\mu \geq 0$ (longueur de la queue) et $\lambda \geq 1$ (longueur du cycle) telles que :*

- *$x_0, x_1, \ldots, x_{\mu-1}$ n'apparaissent qu'une fois dans la suite.*
- *$x_0, x_1, \ldots, x_{\mu+\lambda-1}$ sont tous distincts.*
- *pour $n \geq \mu$, on a $x_{n+\lambda} = x_n$.*

Lorsque $\mu = 0$, on dit que la suite est purement périodique.

La démonstration de ce théorème repose sur le principe des tiroirs rappelé plus haut dans le paragraphe sur le paradoxe des anniversaires. Les éléments $x_0, x_1, \ldots, x_m$, sont en nombre $m + 1$, et sont tous dans l'ensemble E qui a m éléments. Il y en a donc au moins deux qui sont égaux. Maintenant, parmi ces couples $\{i, j\}$, $0 \leq i < j \leq m$ tels que $x_i = x_j$, on choisit ceux tels que i est minimum et, parmi ces derniers, celui tel que j soit minimum, et on appelle ce couple $\{\mu, \mu + \lambda\}$.

On a $x_\mu = x_{\lambda+\mu}$ par construction et

$$x_{\mu+1} = f(x_\mu) = f(x_{\lambda+\mu}) = x_{\lambda+\mu+1}$$

puis, par récurrence, $x_{n+\lambda} = x_n$ pour tout $n \geq \mu$.

Montrons que $x_0, x_1, \ldots, x_{\lambda+\mu-1}$ sont tous distincts :

Si l'on avait $x_i = x_j$ avec $0 \leq i < j \leq \lambda + \mu - 1$, on aurait $i \geq \mu$ (puisque μ a été choisi minimum), et donc $x_{i+1} = x_{j+1}$ et pour tout $t \geq 1$, $x_{i+t} = x_{j+t}$.

En faisant $t = \lambda + \mu - j \geq 1$, on aurait :

$$x_{i+t} = x_{\lambda+\mu} = x_\mu$$

et

$$\mu + 1 \leq i + t = \lambda + \mu + i - j < \lambda + \mu$$

et cela contredirait le choix de $\lambda + \mu$ comme indice minimum.

Il reste à montrer que $x_0, x_1, \ldots, x_{\mu-1}$ n'apparaissent qu'une fois dans la suite : les valeurs prises par x_n pour $n \geq \mu$ sont exactement les valeurs $x_\mu, x_{\mu+1}, x_{\lambda+\mu-1}$ et ces valeurs sont distinctes de $x_0, x_1, \ldots, x_{\mu-1}$.

Épacte. Avec les notations du théorème de la poële à frire, nous allons montrer qu'il existe des indices n tels que $x_n = x_{2n}$. Le plus petit de

ces indices est appelé épacte de la suite (x_n). Il est noté e, et vérifie que $e = \lambda$ si $\mu = 0$, et si $\mu \geq 1$,

$$\mu \leq e \leq \mu + \lambda - 1.$$

D'après le théorème ci-dessus, pour avoir $x_i = x_j$ avec $i < j$, il faut avoir $i \geq \mu$ et $j - i$ multiple de λ. Pour avoir $x_n = x_{2n}$, il faut avoir $n \geq 1$, $n \geq \mu$, et $2n - n = n$ multiple de λ.

Si $\mu = 0$, n est le plus petit multiple de λ strictement positif, c'est donc λ. Si $\mu \geq 1$, e est le plus petit multiple de λ qui est $\geq \mu$: c'est donc l'un des nombres $\mu, \mu + 1, \ldots, \mu + \lambda - 1$.

La méthode de factorisation « ro » de Pollard

Cette méthode est née de la question suivante, complètement indépendante de l'arithmétique : si dans le théorème de la poële à frire, on choisit au hasard le premier terme de la suite x_0, et l'application f de E dans E, quelles sont, en moyenne, la taille de la longueur de la queue μ, de celle du cycle λ, et de l'épacte e ? Les seules inégalités fournies par la démonstration de ce théorème sont $0 \leq \mu \leq m - 1$, $\lambda \geq 1$, $\mu + \lambda \leq m$, et $\mu \leq e \leq \mu + \lambda$. On pourrait donc s'attendre à ce que les valeurs moyennes de μ, λ, e soient de l'ordre de grandeur de m, lorsque m tend vers l'infini.

En fait, par le paradoxe des anniversaires, les égalités $x_i = x_j$ de deux termes de la suite arrivent beaucoup plus souvent que l'on pourrait le penser, et la taille moyenne de μ, λ, e est voisine de $\sqrt{m}$ (de la même façon que 23 est plus proche de $\sqrt{365}$ que de 365).

Plus mathématiquement, si l'on considère comme également probables les m éléments de E comme premier terme de la suite, et qu'aussi les m^m applications de E dans E sont également probables, alors on peut montrer que

$$\text{valeur moyenne de } \mu = \text{valeur moyenne de } \lambda \sim \sqrt{\frac{\pi m}{8}} = 0{,}62\ldots\ \sqrt{m}.$$

et que (c'est un peu plus difficile) :

$$\text{valeur moyenne de } e \sim \sqrt{\frac{\pi^5 m}{288}} = 1{,}03\ldots\ m.$$

Soit c un nombre qui vaut soit + 1, soit − 1, et p un nombre premier.

On choisit $E = \{0, 1, \ldots, p-1\}$, $x_0 = 2$, et $f(x) = x^2 + c \bmod p$.

Lorsque $p = 17$ et $c = -1$, c'est l'exemple que nous avions choisi pour illustrer le théorème de la poële à frire. On construit la suite $x_1 = f(x_0), \ldots, x_{n+1} = f(x_n)$ et on désigne par $e(p, c)$ l'épacte de cette suite.

Conjecture : Pour $c = \pm 1$, et tout p premier, on a :

$$e(p, c) \leq \frac{4}{3}\sqrt{p \log p}.$$

Cette conjecture a été vérifiée par ordinateur pour tout p inférieur à un million. Pour tout $p < 10^6$, on a $e(p, c) \leq 3\,800$ pour $c = \pm 1$.

Compte tenu du raisonnement probabiliste précédent, cette conjecture paraît assez raisonnable. Cependant, ce raisonnement probabiliste ne permet pas de démontrer quoi que ce soit pour l'épacte d'une suite particulière, avec c et p fixés, et on ne voit pas du tout comment attaquer cette conjecture.

Soit N un nombre à factoriser. On s'assure d'abord que N n'est pas premier (par exemple en utilisant le test de Fermat décrit plus haut), puis on réalise l'algorithme suivant :

Données X0 = 2 ; $c = \pm 1$; N à factoriser.

X : = X0 ;
Y : = X0 ;
Pour k de 1 à $k$max faire
X : = $X^2 + c \bmod N$;
Y : = $Y^2 + c \bmod N$;
Y : = $Y^2 + c \bmod N$;
D : = pgcd (Y − X, N) ;
Si D ≠ 1, s'arrêter ;
k suivant.

Soit z_k la suite de nombres entiers naturels définis par $z_0 = 2$, et $z_{k+1} = z_k^2 + c$. On remarque que, dans la boucle de l'algorithme ci-dessus, juste avant le calcul de D, la mémoire X contient $z_k \bmod N$ et Y contient $z_{2k} \bmod N$.

Soit p un facteur premier de N, et soit $e = e(p, c)$ l'épacte de la suite x_k définie précédemment, et qui est exactement :

$$x_k = z_k \bmod p.$$

Que se passe-t-il lorsque $k = e$ dans l'algorithme ci-dessus ?

On a $x_k = x_{2k}$ par définition de l'épacte, ce qui se traduit par p divise $z_{2k} - z_k$. Comme p divise N, p divise Y − X qui vaut $z_{2k} - z_k \bmod N$, et ainsi p divise D qui sera différent de 1. L'algorithme s'arrêtera donc sûrement lorsque $k = e = e(p, c)$.

Si N n'est pas premier, son plus petit facteur premier p est $\leq \sqrt{N}$, et si la conjecture ci-dessus est admise, on aura :

$$e(p, c) \leq \frac{4}{3}\sqrt{p \log p} \leq \frac{4}{3\sqrt{2}} N^{1/4} \sqrt{\log N}$$

ce qui est nettement plus rapide que l'algorithme des divisions successives par les nombres premiers $\leq \sqrt{N}$.

Remarques

1. En général, lorsque l'algorithme s'arrête, D est un diviseur de N différent de N lui-même. On vérifie alors que D et N/D sont des nombres premiers, auquel cas la factorisation est terminée. Si D ou N/D ne sont pas premiers, on recommence avec eux la méthode « ro » de factorisation. Il peut arriver que D soit égal à N. C'est le cas avec N = 1 591 = 7 x 43 et $c = -1$, car les épactes $e(7, -1)$ et $e(43, -1)$ sont toutes deux égales à 6. Dans ce cas, on recommence la méthode ro avec une autre valeur de c.

2. La méthode n'est absolument pas démontrée, cependant, on constate qu'elle marche, et qu'elle fournit un résultat D dont il est facile de voir que c'est un diviseur de N. Les explications précédentes ne sont pas une preuve, mais des observations qui ont conduit les auteurs de la méthode à penser qu'elle serait efficace.

3. Si l'on exécute l'algorithme avec $k$max = 3 800, et que toutes les valeurs de D calculées sont égales à 1, alors on est sûr que le nombre N n'a pas de facteurs premiers ≤ 1 000 000. Cela résulte des calculs de $e(p, c)$ pour p ≤ 1 000 000 et $c = \pm 1$ qui ont montré que $e(p, c)$ est toujours ≤ 3 800. C'est une très bonne méthode pour s'assurer qu'un nombre n'a pas de petits facteurs premiers.

4. Pourquoi $f(x) = x^2 + c$? D'abord on observe que sur l'ensemble $E = \{0, 1, \ldots, p-1\}$, toute fonction de E dans E peut s'écrire $f(x) \equiv P(x) \bmod p$, où $P(x)$ est un polynôme de degré $\leq p - 1$. (En fait, $f(x)$ est égal à son polynôme d'interpolation de Lagrange). Si $f(x)$ est un polynôme du premier degré, l'épacte de la suite $x_{n+1} = f(x_n)$ peut être assez grande, et ce phénomène est utilisé dans la construction de générateurs de nombres au hasard. On choisit donc un polynôme de degré 2 (mais la méthode marcherait aussi en choisissant $f(x) = x^3 + 1$, seulement les calculs seraient un peu plus longs). Le choix de $c = 0$ n'est pas judicieux. Le choix de $c = -2$ non plus pour des raisons plus longues à expliquer. Ces deux valeurs donnent une épacte qui peut être grande. Tout autre valeur de c peut être utilisée.

5. Lorsque l'on fait une analyse d'urines sur une population à faible risque, on peut mélanger les urines de 10 personnes et faire un test sur cet échantillon global. Si le test est négatif, les 10 personnes sont saines. Si le test global se révèle positif, il faut refaire des tests individuels. Dans

l'algorithme ci-dessus, le coût du calcul du pgcd est assez élevé. On peut initialiser A à 1 au tout début, et remplacer le calcul de D par :

$$A := A * (Y - X) \bmod N$$

Si k multiple de 10, alors

D : = pgcd (A, N) ;

Si D ≠ 1, s'arrêter ;

A : = 1.

Il faut tester si p divise (Y – X). On ne le fait qu'une fois sur 10, en multipliant entre elles les valeurs successives de Y – X. Comme dans les analyses d'urines, le choix de 10 n'est pas forcément le meilleur.

6. La méthode est très facile à programmer. Elle nécessite un langage d'ordinateur muni de multiprécision, c'est-à-dire d'un logiciel traitant les grands entiers, si on ne veut pas se cantonner à des nombres trop petits. Mais ces logiciels sont de plus en plus fréquents.

Jean-Louis NICOLAS
Janvier 1994