Algorithmes arithmétiques classiques

Les applications de l'arithmétique et le développement des moyens de calcul ont rendu cruciales les questions sur la possibilité effective de réaliser certaines opérations. Prenons par exemple un nombre n produit de deux grands nombres premiers p et q de 150 chiffres décimaux chacun. Peut-on calculer p et q connaissant n ? Bien entendu il existe un algorithme naïf : essayer de diviser n successivement par tous les nombres qui lui sont inférieurs. Mais compte tenu de la taille des nombres utilisés, cet algorithme ne peut se réaliser en un temps raisonnable. Jusqu'à présent, bien que des progrès soient faits sur les algorithmes utilisés et sur la puissance de calcul des machines, on ne sait pas résoudre pratiquement un tel problème. En revanche si a et b sont deux nombres très grands, on sait trouver en temps raisonnable $d = pgcd(a, b)$ ainsi que u et v tels que $au + bv = d$.
Ainsi, **il y a des opérations réalisables pratiquement et d'autres qui dans l'état actuel de nos connaissances et de nos moyens techniques ne le sont pas**. Cet état de fait est utilisé pour construire des systèmes cryptographiques qu'on espère robustes.

1 Division euclidienne

Le premier résultat fondamental de l'arithmétique est l'existence d'une **division euclidienne**.

Théorème 1.1 *Soient a et b deux entiers. Nous supposons b non nul. Alors il existe un et un seul couple d'entiers (q, r) tels que*

$$a = bq + r \ \ et \ \ 0 \leq r < |b|.$$

1.1 Algorithme de division euclidienne par soustraction

$B := b;$
$R := a;$
$Q := 0;$
tant que $R \geq B$ *faire*
  début
    $R := R - B$;
    $Q := Q + 1$;
  fin ;

1.2 Version binaire de l'algorithme de division euclidienne

On calcule avant toute chose par duplications successives le plus petit entier $n \geq 0$ tel que $2^n b > a$.

$B := b;$
$R := a;$
$Q := 0;$
$N := n;$
$Aux := 2^N B;$
tant que $N > 0$ *faire*
  début
    $Aux := Aux/2;$
    $N := N - 1;$
    si $R < Aux$
      alors $Q := 2 * Q$
      sinon début
        $Q := 2 * Q + 1;$
        $R := R - Aux;$
      fin ;
  fin ;

Remarquons que les deux algorithmes proposés s'exécutent en des temps qui ne sont pas du même ordre. Si on fixe b, le premier demande un nombre de tours de boucles de l'ordre de a, donc **exponentiel** en la taille de a (la taille de a, nombre de bits nécessaires pour représenter a, est de l'ordre de $\log_2(a)$ bits). Le deuxième demande tout d'abord de déterminer le plus petit entier n tel que $2^n b > a$. Ceci se fait en temps polynomial. Avec cette valeur de n, le nombre de tours de boucles effectué est de l'ordre de $\log_2(a)$, donc **linéaire** en la taille de a. Dans les deux cas, chaque boucle ne contient bien entendu que des opérations élémentaires.
La division euclidienne dans $\mathbb{Z}$ est une **relation d'ordre**. Une partie de l'arithmétique repose sur l'étude de cette relation de divisibilité. En particulier il est utile d'introduire la borne inférieure de deux éléments (**plus grand commun diviseur**) et la borne supérieure (**plus petit commun multiple**).

Théorème 1.2 *Soient a et b deux entiers dont l'un au moins est non nul. Il existe un plus grand entier > 0 qui soit diviseur commun de a et de b. Cet entier noté $pgcd(a, b)$ est appelé le* **plus grand commun diviseur** *de a et b.*

1.3 Algorithme d'Euclide

$R0 := |a|;$
$R1 := |b|; \quad (b \neq 0)$
tant que $R1 > 0$ *faire*
  début
    $R := Reste\_Division(R0, R1)$;
    $R0 := R1$;
    $R1 := R$;
  fin ;

en sortie $R1 = 0$, et $R0 = pgcd(a, b)$.

Le résultat qui suit, appelé théorème de Bézout, est très important. Nous verrons qu'il est lié au **calcul de l'inverse** dans les classes résiduelles modulo un entier n.

Théorème 1.3 *Si $pgcd(a, b) = d$, il existe deux entiers u et v tels que $ua + vb = d$.*

1.4 Algorithme d'Euclide étendu

Là encore nous supposerons que $a \geq 0$ et $b > 0$. Le cas général s'en déduit.
Voici un algorithme (**algorithme d'Euclide étendu**, adaptation de l'algorithme précédent) qui permet de trouver explicitement un couple (u, v) qui convient.

$R0 := a; \quad (a \geq 0)$
$R1 := b; \quad (b > 0)$
$U0 := 1; U1 := 0;$
$V0 := 0; V1 := 1;$
tant que $R1 > 0$ *faire*
  début
    $Q := Quotient\_Division(R0, R1)$;
    $R := Reste\_Division(R0, R1)$;
    $U := U0 - Q * U1$;
    $V := V0 - Q * V1$;
    $R0 := R1$; $R1 := R$;
    $U0 := U1$; $U1 := U$;
    $V0 := V1$; $V1 := V$;
  fin ;

Remarquons qu'il s'agit d'une amélioration de l'algorithme d'Euclide donné précédemment pour le calcul du $pgcd$. Comme précédemment

l'algorithme se termine avec $R1 = 0$ et $R0 = pgcd(a, b)$.

1.5 Complexité de l'algorithme d'Euclide et de l'algorithme d'Euclide étendu

Le nombre de pas de l'algorithme d'Euclide est donc majoré par une fonction linéaire de la taille de l'entrée.

2 Les classes résiduelles

2.1 Les éléments inversibles de $\mathbb{Z}/n\mathbb{Z}$

Rappelons les notations usuelles concernant les congruences. On dit que x est congru à y modulo n et on note

$$x \equiv y \quad (n),$$

lorsque $x-y$ est un multiple de n, c'est-à-dire lorsqu'il existe un entier k tel que

$$x = y + kn.$$

La congruence est une relation d'équivalence et l'ensemble des classes d'équivalence est noté $\mathbb{Z}/n\mathbb{Z}$. Dans chaque classe il y a un représentant x et un seul tel que $0 \leq x < n$. Ainsi on peut considérer que les éléments de $\mathbb{Z}/n\mathbb{Z}$ sont $0, 1, 2, \cdots, n-1$. L'addition et la multiplication dans $\mathbb{Z}/n\mathbb{Z}$ se font en additionnant et en multipliant dans $\mathbb{Z}$ puis en réduisant modulo n. On notera

$$x = y \ \ mod \ \ n,$$

l'unique élément x congru à y modulo n et tel que $0 \leq x < n$ (attention dans de nombreux langages informatiques la fonction mod ne renvoie pas tout à fait cela quand le nombre y est négatif).

Théorème 2.1 *La classe de x est inversible dans $\mathbb{Z}/n\mathbb{Z}$ si et seulement si x est premier avec n.*

Preuve. Le point important est de voir le lien qu'il y a entre l'inversibilité dans $\mathbb{Z}/n\mathbb{Z}$ et le théorème de Bézout. La classe de x est inversible dans $\mathbb{Z}/n\mathbb{Z}$ si et seulement s'il existe u et v tels que $ux = 1+vn$. Cette dernière relation est une relation de Bézout. On sait qu'elle est vérifiée et seulement si x et n sont premiers entre eux. □

Théorème 2.2 *L'anneau $\mathbb{Z}/n\mathbb{Z}$ est un corps si et seulement si n est premier.*

Preuve. Pour que l'anneau $\mathbb{Z}/n\mathbb{Z}$ soit un corps il faut et il suffit que tout élément non nul soit inversible. Par application du théorème précédent ceci se produit si et seulement si n est un nombre premier.

Théorème 2.3 *Si m et n sont premiers entre eux alors la condition*

$$\begin{cases} a \equiv b & (m) \\ a \equiv b & (n) \end{cases}$$

est équivalente à

$$a \equiv b \quad (mn).$$

Soient m et n premiers entre eux. On cherche toutes les solutions entières de

$$\begin{cases} x \equiv a & (m) \\ x \equiv b & (n) \end{cases}$$

On considére u et v tels que $um + vn = 1$.

Théorème 2.4 *(Théorème des restes chinois) On obtient une solution en prenant*

$$x = bum + avn$$

Toutes les solutions sont alors de la forme

$$x + kmn.$$

Théorème 2.5 *Soit p un nombre premier. Si $1 \leq k \leq p-1$ alors*

$$\binom{p}{k} \equiv 0 \quad (p).$$

Théorème 2.6 *(Petit théorème de Fermat) Si a est premier avec p alors*

$$a^{p-1} \equiv 1 \quad (p).$$

Théorème 2.7 *Soit $n = pq$ où p et q sont deux nombres premiers. Alors, pour tout a et tout k on a*

$$a^{k(p-1)(q-1)+1} \equiv a \quad (n).$$

La démonstration que nous venons de faire s'applique en remplaçant le produit $(p-1)(q-1)$ par le plus petit commun multiple $ppcm(p-1, q-1)$. En conséquence nous obtenons :

Théorème 2.8 *Soit $n = pq$ où p et q sont deux nombres premiers. Alors, pour tout a et tout k on a*

$$a^{k\, ppcm(p-1,q-1)+1} \equiv a \quad (n).$$

2.3 La fonction d'Euler

Notons $(\mathbb{Z}/n\mathbb{Z})^*$ le groupe multiplicatif des éléments inversibles de $\mathbb{Z}/n\mathbb{Z}$. Nous noterons $\Phi(n)$ le nombre des éléments de $(\mathbb{Z}/n\mathbb{Z})^*$. On posera en outre $\Phi(1) = 1$. (fonction d'Euler).

Lemme 2.9 *Si p est premier alors*

$$\Phi(p) = p - 1.$$

Lemme 2.10 *Si p est un nombre premier et α un entier ≥ 1 alors*

$$\Phi(p^\alpha) = (p-1)p^{\alpha-1}.$$

Lemme 2.11 *Si m et n sont premiers entre eux alors*

$$\Phi(mn) = \Phi(m)\Phi(n).$$

Théorème 2.12 *Pour tout $n > 1$*

$$\Phi(n) = n \prod_i \left(1 - \frac{1}{p_i}\right),$$

où les p_i sont les facteurs premiers de n.

Théorème 2.13 *Soit a un élément de $(\mathbb{Z}/n\mathbb{Z})^*$ (c'est-à-dire a premier avec n) alors*

$$a^{\Phi(n)} \equiv 1 \quad (n).$$

2.4 Calcul d'une puissance

Soit $n \geq 1$ un entier. On veut calculer a^n (a est par exemple dans $\mathbb{Z}/n\mathbb{Z}$ ou dans $\mathbb{R}$, $\mathbb{C}$,...). On considère l'algorithme suivant

$A := a;$
$N := n;$
$R := 1;$
tant que $N > 0$ *faire*
  si N *pair*
    alors début
      $A := A * A;$
      $N := N/2;$
    fin
    sinon début
      $R := R * A;$
      $N := N - 1;$
    fin.

cet algorithme se termine et en sortie R contient a^n.

Dans le pire des cas, la valeur de N est divisée par 2 toutes les 2 étapes de l'algorithme. Ainsi le nombre de tours de boucle est $O(\log(n))$, c'est-à-dire linéaire par rapport à la taille de n. Chaque tour de boucle ne contient que des opérations élémentaires. Remarquons que lorsqu'on travaille modulo n on réduit modulo n à chaque tour de boucle (sinon les calculs intermédiaires risquent de faire intervenir des nombres énormes).

Soit a un élément inversible de $\mathbb{Z}/n\mathbb{Z}$. On sait alors que

$$a^{\Phi(n)} \equiv 1 \quad (n).$$

Donc

$$a^{-1} \mod n = a^{\Phi(n)-1} \mod n.$$

On dispose donc de deux algorithmes pour calculer l'inverse dans $\mathbb{Z}/n\mathbb{Z}$:

- de l'algorithme d'Euclide étendu ;
- de l'algorithme de calcul d'une puissance conjugué avec l'égalité précédente.

Grossièrement nous avons vu que ces deux algorithmes utilisent un nombre de tours de boucle linéaire en la taille de n. Nous sommes dans le cas où un calcul du nombre d'itérations ne permet pas de comparer l'efficacité de ces deux algorithmes. Pour les comparer plus finement il faudrait faire une analyse au niveau des opérations sur les bits (ou les octets) du coût de chacun. Il est laissé au lecteur le soin de faire cette comparaison. Dans un modèle réaliste, l'algorithme d'Euclide étendu s'avère plus avantageux.

Compte tenu de la remarque précédente sur les coûts, voici un exemple intéressant. Il s'agit de calculer $a^m \mod n$ où $n = pq$, quand on connaît la factorisation pq.

Une première idée est d'appliquer brutalement l'algorithme de calcul d'une puissance qu'on vient de décrire.

Une méthode un peu plus rapide consiste à faire la démarche suivante. On calcule $x = a^m \mod p$ et $y = a^m \mod q$ et on reconstitue $z = a^m$

$\mod n$ par application du théorème des restes chinois. Pour cela on calcule u et v tels que $up + vq = 1$ et on en déduit $z = uyp + vxq \mod n$.
Pratiquement on peut procéder de la façon suivante. on calcule v puis x et y. On calcule

$$h = (x - y)v \mod p.$$

On a alors

$$z = y + qh.$$

En effet il est facile de voir que $(y + qh) \mod q = y$, que $(y + qh) \mod p = x$ et que $0 \leq y + qh < n$.
Dans cette méthode, les nombres manipulés dans les calculs intermédiaires sont plus petits puisqu'on travaille modulo p et modulo q. On peut donc s'attendre à un meilleur temps d'exécution. Un évaluation plus fine, confirmée par l'expérience, montre que cet algorithme est à peu près 4 fois plus rapide que l'application directe de l'algorithme de calcul d'une puissance.

2.5 Générateurs de $(\mathbb{Z}/p\mathbb{Z})^*$ (p premier) - Logarithme discret

Dans toute la suite p est un nombre premier et $(\mathbb{Z}/p\mathbb{Z})^*$ est le groupe multiplicatif des éléments non nuls du corps $\mathbb{Z}/p\mathbb{Z}$.
On rappelle que d'après le petit théorème de Fermat pour tout $a \in (\mathbb{Z}/p\mathbb{Z})^*$

$$a^{p-1} = 1.$$

Soit $a \in (\mathbb{Z}/p\mathbb{Z})^*$, il existe donc un plus petit entier $e > 0$ tel que $a^e = 1$.
L'entier e est appelé l'ordre de a.

Théorème 2.14 *L'ordre e d'un élément a de $(\mathbb{Z}/p\mathbb{Z})^*$ divise tout élément s tel que $a^s = 1$. En particulier e divise $p - 1$.*

Théorème 2.15 *Si e_a est l'ordre de a, si e_b est l'ordre de b et si e_a et e_b sont premiers entre eux, alors l'ordre de ab est $e_a e_b$.*

Théorème 2.16 *Le plus petit commun multiple des ordres des éléments de* $(\mathbb{Z}/p\mathbb{Z})^*$ *est* $p-1$ *et il existe un élément* α *tel que*

$$(\mathbb{Z}/p\mathbb{Z})^* = \{1, \alpha, \cdots, \alpha^{p-2}\}.$$

En conséquence du lemme précédent il existe un élément α d'ordre $p-1$. $\square$

Définition 2.17 *Les éléments qui ont cette propriété sont appelés* **éléments primitifs**.

Lemme 2.18 *Si* a *est d'ordre* e_a*, tout autre élément d'ordre* e_a *est nécessairement de la forme* a^i *avec* $1 \leq i \leq e-1$.

Théorème 2.19 *L'ordre de* a^i *est* $ppcm(i, e_a)/i$.

Corollaire 2.20 *Si* a *est un élément d'ordre* e_a *alors* a^i *est d'ordre* e_a *si et seulement si* i *est premier avec* e_a*. Il y a* $\Phi(e_a)$ *éléments d'ordre* e_a.

Théorème 2.21 *Pour tout diviseur* e *de* $p-1$ *il y a exactement* $\Phi(e)$ *éléments d'ordre* e.

Voici un algorithme pour trouver un élément primitif. On décompose $p-1$ en facteurs premiers

$$p-1 = q_1^{\alpha_1} \cdots q_k^{\alpha_k}.$$

On choisit un a et on calcule pour tout i

$$a^{\frac{p-1}{q_i}}.$$

Si tous ces calculs donnent des résultats tous distincts de 1 alors a est un élément primitif, sinon on choisit un autre a. La probabilité de succès pour un tirage d'un élément a est $\frac{\Phi(p-1)}{p-1}$.
Cependant, cet algorithme suppose qu'on connaisse la décomposition de $p-1$ en facteurs premiers. Ceci n'est pas réalisable en pratique pour des grands nombres sauf si par exemple on connaît à l'avance un grand facteur premier q de $p-1$ de telle sorte que $(p-1)/q$ soit suffisamment petit pour pouvoir être factorisé.

Définition 2.22 *Soit α un élément primitif de $(\mathbb{Z}/p\mathbb{Z})^*$. Tout élément a de $(\mathbb{Z}/p\mathbb{Z})^*$ s'écrit donc*

$$a = \alpha^k.$$

Le nombre k est le **logarithme discret** *de a à base α.*

2.6 Quelques résultats sur les logarithmes discrets

La notion de logarithme discret donne naissance à un problème réputé difficile : le **problème du logarithme discret**. Ce problème est le suivant : on se place dans le groupe multiplicatif cyclique $(\mathbb{Z}/p\mathbb{Z})^*$ dont on fixe un élément primitif α ; soit x un élément de $(\mathbb{Z}/p\mathbb{Z})^*$: trouver k tel que $\alpha^k = x$.
Nous renvoyons à [**?**] (5.3, 6.4) et à [**?**] (5.1.1) pour une étude des algorithmes de calcul des logarithmes discrets. Les meilleurs algorithmes actuels dans le cas du groupe $(\mathbb{Z}/p\mathbb{Z})$ sont sous-exponentiels (méthodes de l'index).

3 Nombres premiers

3.1 Déterminer si un nombre est premier ou non

Pouvoir engendrer de grands nombres premiers s'avère être capital dans de nombreux systèmes cryptographiques. Pour un n donné, on ne connaît pas d'algorithme générique permettant de construire un nombre premier aléatoire inférieur ou égal à n. Cependant un célèbre résultat de la théorie des nombres dû à J. Hadamard et C.J. de la Vallée Poussin, stipule que pour n grand, $\sharp\{p \text{ premier} \mid p \leq n\} \sim \frac{n}{\ln(n)}$. Ainsi la méthode couramment utilisée pour construire un nombre premier $p \leq n$ est de tirer aléatoirement un entier de cet intervalle et de l'incrémenter jusqu'à obtenir un nombre premier. Il faut donc disposer d'algorithmes efficaces permettant de vérifier si un nombre est premier ou non. En 1975, V. Pratt a démontré que ce problème était dans la classe NP. Le problème complémentaire (vérifier si un nombre

peut se décomposer en un produit de plusieurs facteurs) est évidemment dans NP aussi. En 2002, M. Agrawal, N. Kayal et N. Saxena ont démontré que le problème de la primalité est en fait un **problème polynomial**. Ce résultat théorique extrêmement intéressant (exhibant au passage l'un des rares problèmes connus à être dans $P \cap co - NP$) n'est malheureusement pas praticable.
On dispose essentiellement de deux types d'algorithmes utilisables pour vérifier si un nombre est premier. Tout d'abord des tests (par exemple le **test de Miller-Rabin**) de non primalité qui comme leur nom l'indique peuvent certifier qu'un nombre est non premier, c'est-à-dire que si le test répond oui, alors on est sûr que le nombre est composé. En revanche si le test répond non alors il est vraisemblablement premier, sans que ce soit une certitude. On a de cette façon des algorithmes de Monte Carlo. Comme on peut rendre la probabilité de se tromper, lorsqu'on décrète avec un tel test qu'un nombre est premier, très petite (par exemple $< 1/2^{100}$) on peut souvent s'en contenter. Si toutefois on veut une preuve absolue on peut employer un algorithme qui donne une certitude, par exemple l'algorithme de Adleman, Pomerance, Rumely, Cohen, Lenstra (dénommé algorithme APRCL), et qui quoique plus lent est tout de même praticable sur des nombres de 1000 décimales et dont le temps d'exécution est en $O(t^{C \ln \ln(t)})$ où t est la taille de l'entier n qu'on étudie. Bien entendu on n'utilise un tel algorithme qu'après avoir détecté qu'un nombre est probablement premier.

4 Algorithmes de construction de nombres premiers

Nous donnons ici quelques méthodes effectives de construction de nombres premiers soumis à certaines conditions. Ces algorithmes sont très utiles dans l'implémentation effective de nombreux systèmes cryptographiques de chiffrement, de signature, d'échange de clés ; par exemple : ElGamal, RSA, DSA, Diffie-Hellman.

4.1 Construire un grand nombre premier au hasard

Construire un grand nombre premier au hasard est simple et rapide. Supposons qu'on veuille un nombre premier de 1024 bits. On tire au sort un nombre impair de 1024 bits. On teste avec le test de Miller-Rabin s'il est vraisemblablement premier. Si c'est le cas on peut s'en contenter ou utiliser l'algorithme APRCL pour en être sûr. S'il n'est pas premier on ajoute 2 et on réitère le procédé. L'espérance du nombre d'itérations nécessaire est obtenu grâce au théorème des nombres premiers, conjecturé par Gauss et Legendre et prouvé par Hadamard et de la Vallée Poussin :

Théorème 4.1 *Soit*

$$\pi(x) = |\{p\ premier \mid p \leq x\}|.$$

Alors pour x grand on a

$$\pi(x) \sim \frac{x}{\ln(x)}.$$

Le nombre moyen d'itérations nécessaires est donc de l'ordre de la taille de x.

4.2 Construire un couple (p, q) de nombres premiers tels que q divise $p - 1$

Supposons qu'on veuille construire un nombre premier p de 1024 bits, tel que $p - 1$ ait un grand facteur premier q de 874 bits par exemple. C'est ce qu'on est amené à faire quand on veut trouver un élément primitif de $\mathbb{Z}/p\mathbb{Z}$. On procède alors de la façon suivante : on construit au hasard un nombre premier q de 874 bits comme indiqué précédemment. Puis on tire au hasard un nombre pair $2k$ ayant 150 bits et on calcule $2kq + 1$. Si c'est un nombre premier on a terminé, sinon on remplace k par $k + 1$ (c'est-à-dire on rajoute $2q$ au nombre précédent). Compte tenu du théorème de Dirichlet suivant (cf. le théorème 4.2), on voit que cet algorithme aboutit en un temps moyen de l'ordre de la taille désignée pour p.

Théorème 4.2 *Soit*

$$\pi_{q,c}(x) = |\{p\ premier \mid p \leq x, p = kq + c\}|.$$

Alors pour x grand on a

$$\pi_{q,c}(x) \sim \frac{1}{\phi(q)} \frac{x}{\ln(x)}.$$

On applique dans le cas qui nous occupe ce théorème avec $c = 1$. Il y a alors moins de $x/2q$ valeurs de $2k$ telles que $2kq+1 \leq x$. Parmi celles-ci il y en a de l'ordre de $\pi_{q,1}(x)$ qui conviennent, donc une proportion de l'ordre de

$$\frac{2q}{(q-1)} \frac{1}{\ln(x)}.$$

L'espérance du nombre d'itérations à faire est donc de l'ordre de $\ln(x)$, c'est-à-dire de la taille désignée pour p.

Remarque. si lors de la phase d'itération, la taille de k vient à dépasser 149 bits on recommence avec une nouvelle valeur de q.

4.3 Construire un couple (p, q) de nombres premiers tels $p = 2q + 1$

Ce problème se pose en cryptographie, notamment dans le cadre du chiffrement d'ElGamal. Le nombre premier q est alors appelé nombre premier de Sophie Germain. La méthode de construction est encore la même, on tire q au sort et on regarde si $p = 2q + 1$ est premier. Cependant ici on ne dispose d'aucun résultat théorique pour affirmer qu'on va arriver à construire un tel couple (p, q) en temps raisonnable. En effet, on ne sait pas grand chose sur les nombres de Sophie Germain. On ne sait pas s'il y en a une infinité. On conjecture cependant que le nombre de nombres premiers de Sophie Germain $\leq n$ est asymptotiquement équivalent à

$$\frac{2Cn}{(\ln(n))^2}$$

où C est une constante voisine de 0.66.

5 Résidus quadratiques

Le problème posé ici est celui de la racine carrée d'un entier modulo n. Plus précisément :

Définition 5.1 *Soient a et n deux entiers premiers entre eux. Nous dirons que a est un résidu quadratique modulo n si l'équation $x^2 \equiv a \quad (n)$ a des solutions.*

Ce problème se décompose en au moins deux questions principales.
- Déterminer si un nombre est un résidu quadratique modulo n.
- Si un nombre est un résidu quadratique, déterminer une racine carrée et plus généralement les trouver toutes.

Nous allons dans un premier temps travailler avec un module premier. Si ce module est 2 le problème est très simple. En fait le problème se pose avec de grands nombres premiers, qui sont donc impairs. Ensuite nous traiterons le cas où le module est un produit de deux grands nombres premiers.

5.1 Cas d'un module premier impair de la forme $4k-1$: cas simple

Soit p un nombre premier de la forme $4k-1$. Soit $n = a^2$ dans $\mathbb{Z}/p\mathbb{Z}$. On vérifie par un calcul simple que $n^{\frac{p+1}{4}}$ est une racine carrée de n.

5.2 Cas d'un module premier impair de la forme $4k+1$: algorithme de Shank

On suppose maintenant que p est de la forme $4k+1$. On écrit alors

$$p - 1 = 2^s t$$

avec t impair et $s \geq 2$.
Soit $n = a^2$ dans $\mathbb{Z}/p\mathbb{Z}$. On suppose qu'on connaît un m qui ne soit pas un résidu quadratique modulo p. Posons alors

$$z = m^t.$$

Par suite :

$$z^{2^{s-1}} = m^{t2^{s-1}} = m^{\frac{p-1}{2}}.$$

Comme m n'est pas un carré on obtient :

$$z^{2^{s-1}} \equiv -1 \quad (p).$$

Posons

$$B = n^t, \quad X = n^{\frac{t+1}{2}}, \quad Y = z, \quad R = s - 1$$

et effectuons l'algorithme suivant :

tant que $R \geq 1$ *faire*
  début
    si $B^{2^{R-1}} \equiv 1 \ (p)$
      alors
        début
          $Y = Y^2$;
          $R = R - 1$;
        fin
      sinon
        début
          $B = BY^2$;
          $X = XY$;
          $Y = Y^2$;
          $R = R - 1$;
        fin
  fin ;

On vérifie qu'en sortie, X contient une racine carrée de n.
Remarquons que cet algorithme donne le résultat pourvu qu'on ait tiré au sort au début un m qui ne soit pas un résidu quadratique. Ceci donne naissance à un algorithme de Las Vegas.

5.3 Cas d'un module produit de deux grands nombres premiers

On suppose donc que $n = pq$ (p et q premiers) que a est un carré modulo n premier avec n.
Alors a est un carré non nul modulo p et modulo q. Comme on sait chercher des racines carrés modulo un nombre premier, on peut trouver u et v (qui sont aussi non nuls) tels que :

$$u^2 \equiv a \quad (p),$$

$$v^2 \equiv a \quad (q).$$

Pour trouver les racines carrées de a modulo pq on est donc amené à résoudre les 4 systèmes de congruences :

$$\begin{cases} x_1 \equiv u & (p), \\ x_1 \equiv v & (q). \end{cases}$$

$$\begin{cases} x_2 \equiv u & (p), \\ x_2 \equiv -v & (q). \end{cases}$$

$$\begin{cases} x_3 \equiv -u & (p), \\ x_3 \equiv v & (q). \end{cases}$$

$$\begin{cases} x_4 \equiv -u & (p), \\ x_4 \equiv -v & (q). \end{cases}$$

D'après le théorème des restes chinois, nous obtenons une solution (modulo n) pour chaque système. Ces systèmes sont tous distincts puisque u et v ne sont pas nuls. Les solutions x_1, x_2, x_3, x_4 constituent les 4 racines carrées de a modulo n.
Remarque. on peut aussi donner les résultats dans les cas particuliers ; si $a \equiv 0 \quad n$ on a évidemment 0 comme seule racine carrée ; si a est multiple de p ou de q sans être multiple de n, alors on obtient deux racines carrées.

5.4 Symbole de Legendre et symbole de Jacobi

Soit p un nombre premier impair et soit a un entier. Le fait que a soit un carré modulo p ne dépend évidemment que de la classe de a modulo p. Nous allons donc nous placer dans le corps $\mathbb{Z}/p\mathbb{Z}$ et étudier les nombres qui sont des carrés non nuls (autrement dit les résidus quadratiques modulo p). Le groupe multiplicatif $(\mathbb{Z}/p\mathbb{Z})^*$ est cyclique ; soit α un de ses générateurs. Ainsi

$$(\mathbb{Z}/p\mathbb{Z})^* = \{1, \alpha, \alpha^2, \cdots, \alpha^{p-2}\}.$$

Théorème 5.2 *Soit p un nombre premier impair. Un générateur α de $(\mathbb{Z}/p\mathbb{Z})^*$ n'est pas un résidu quadratique. De plus dans $(\mathbb{Z}/p\mathbb{Z})^*$*

$$\alpha^{\frac{p-1}{2}} = -1.$$

Preuve. On sait d'après le petit théorème de Fermat que $\alpha^{p-1} = 1$. On en déduit que $\alpha^{\frac{p-1}{2}} = \pm 1$. Mais comme α est, par définition, d'ordre $p-1$, on a nécessairement $\alpha^{\frac{p-1}{2}} = -1$. Si α était un carré on aurait $\alpha = \beta^2$ et donc on aurait $\alpha^{\frac{p-1}{2}} = \beta^{p-1} = 1$, ce qui n'est pas. □

Théorème 5.3 *Soit p un nombre premier impair et soit α un générateur de $(\mathbb{Z}/p\mathbb{Z})^*$. Dans $(\mathbb{Z}/p\mathbb{Z})^*$ les résidus quadratiques sont les puissances paires de α. Ainsi la moitié des $p-1$ éléments de $(\mathbb{Z}/p\mathbb{Z})^*$ sont des résidus quadratiques.*

Preuve Il est clair qu'une puissance paire α^{2k} de α admet deux racines carrées $\pm\alpha^k$. Si maintenant on prend un nombre de la forme α^{2k+1} il ne peut pas être de la forme u^2. Sinon on aurait $\alpha = \frac{u^2}{\alpha^{2k}}$, c'est-à-dire que α serait un carré, ce qui n'est pas. □

Définition 5.4 *Soit p un nombre premier impair et soit a un entier. Nous définissons le symbole de Legendre de a modulo p par :*

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \textit{si a est divisible par p} \\ 1 & \textit{si a est un résidu quadratique modulo p} \\ -1 & \textit{si a n'est pas divisible par p et n'est pas un résidu quadratiqu} \end{cases}$$

Théorème 5.3 *Si a est premier avec p alors*

$$\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}}.$$

Preuve. Soit α un générateur de $(\mathbb{Z}/p\mathbb{Z})^*$. On peut écrire a sous la forme $a = \alpha^k$. Donc

$$a^{\frac{p-1}{2}} = \left(\alpha^{\frac{p-1}{2}}\right)^k = (-1)^k.$$

Le théorème 5.3 permet de conclure. □

1) Le symbole de Legendre est donc directement lié à la notion de résidu quadratique. Le symbole de Legendre de a modulo p vaut 1 si et seulement si a est un résidu quadratique.

2) Le symbole de Legendre de a ne dépend que de la classe de a :

$$\left(\frac{a}{p}\right) = \left(\frac{a \mod p}{p}\right).$$

3) L'expression du symbole de Legendre donné dans le théorème précédent nous permet de le calculer en temps polynomial, grâce à l'algorithme de calcul de puissance donné dans le paragraphe 2.4.

4) Le symbole de Legendre est multiplicatif :

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right).$$

Nous allons nous intéresser maintenant au cas où le module n n'est pas un nombre premier. Dans ce cas nous allons généraliser le symbole de Legendre en définissant le symbole de Jacobi.

Définition 5.6 *Soit n un entier impair ≥ 3 que nous décomposons sous la forme $n = p_1 p_2 \cdots p_s$ où les p_i sont des nombres premiers (non nécessairement distincts). Soit a un entier. Nous définissons le symbole de Jacobi de a modulo n par :*

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)\left(\frac{a}{p_2}\right)\cdots\left(\frac{a}{p_s}\right).$$

1) Lorsque n est premier impair, le symbole de Jacobi coïncide avec le symbole de Legendre.
2) Le symbole de Jacobi de a ne dépend que de la classe de a :

$$\left(\frac{a}{n}\right) = \left(\frac{a \mod n}{n}\right).$$

3) Le symbole de Jacobi est multiplicatif en ses deux arguments :

$$\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{b}{n}\right),$$

$$\left(\frac{a}{mn}\right) = \left(\frac{a}{m}\right)\left(\frac{a}{n}\right).$$

4) Si a n'est pas premier avec n alors

$$\left(\frac{a}{n}\right) = 0.$$

Si on sait factoriser n, le calcul du symbole de Jacobi passe par le calcul d'un certain nombre de symboles de Legendre. Si n est trop grand pour être factorisé efficacement on peut tout de même calculer le symbole de Jacobi de manière efficace grâce aux deux propositions suivantes.

Théorème 5.7 *Soit $n > 3$ un entier impair. Alors*

$$\left(\frac{-1}{n}\right) = (-1)^{\frac{n-1}{2}},$$

$$\left(\frac{2}{n}\right) = (-1)^{\frac{n^2-1}{8}}.$$

Théorème 5.8 ***(Loi de réciprocité quadratique)*** *Soient m et n des entiers impairs > 3. Alors*

$$\left(\frac{m}{n}\right) = (-1)^{\frac{(n-1)(m-1)}{4}}\left(\frac{n}{m}\right).$$

Nous renvoyons à [?], [?] ou [?] pour une preuve de ces deux théorèmes.

Voici un algorithme polynomial qui permet alors de calculer le symbole de Jacobi. On se donne un entier n impair ≥ 3 et un entier m. On se propose de calculer $\left(\frac{m}{n}\right)$.

$N := n$; $M := M$; $S := 1$;
si $M < 0$
  alors
    début
      $M := -M$;
      $S := (-1)^{\frac{N-1}{2}}$;
    fin ;
tant que $M \geq 2$ *faire*
  si M pair
    alors
      début
        $M := M/2$;
        $S := S * (-1)^{\frac{N^2-1}{8}}$;
      fin
    sinon
      début
        $S := S * (-1)^{\frac{(N-1)(M-1)}{2}}$;
        $Aux := Reste\_Division(N, M)$;
        $N := M$;
        $M := Aux$;
      fin ;
si $M = 0$ alors $S = 0$;

En sortie, S contient le symbole de Jacobi cherché.

Remarque. Le symbole de Jacobi modulo n ne caractérise pas les résidus quadratiques modulo n. Clairement, si a est un résidu quadratique modulo n, alors son symbole de Jacobi est 1. Mais la réciproque est fausse en général. La situation n'est plus aussi simple que dans le cas d'un module premier (symbole de Legendre). Nous venons de voir, grâce à l'algorithme précédent, que le calcul du symbole de Jacobi d'un entier modulo n est polynomial. En revanche, déterminer si un entier est un résidu quadratique ou non modulo un produit $n = pq$ de deux nombres premiers inconnus p et q est un problème qui est pour le moment non polynomial .