

Els primers, manual d'instruccions

Xavier Xarles

Departament de Matemàtiques
Universitat Autònoma de Barcelona
xarles@mat.uab.cat

Resum

En aquest manual expliquem com es fa realment per trobar nombres primers prou grans per ser utilitzats per a la criptografia. També fem una breu incursió en alguns mètodes de factorització.

Abstract

This paper explains exactly how we find the prime numbers large enough to be used in cryptography. It also outlines some methods of factorization.

1. Introducció

Quan ens parlen dels nombres primers, o quan els expliquem a classe, sempre es posa molt d'èmfasi en com són d'útils ara mateix. De fet, crec que no en som realment conscients, de la importància cabdal que tenen i han tingut aquests últims anys. Penseu que cada cop que ens connectem a una pàgina web segura (per exemple, les pàgines que comencen amb «https»), aquesta fa servir un sistema de comunicació xifrada que amb tota seguretat utilitza nombres primers molt grans (en parlarem després, però poden ser de prop d'unes sis-centes xifres decimals). I penseu que això ho feu diverses vegades al dia (probablement, moltes). Ara mateix, centenars de milers d'empreses necessiten utilitzar nombres primers enormes i, a més, que siguin diferents tots ells, i fins i tot anar-los canviant cada poc temps. Així doncs, sembla natural pensar que un mètode per trobar primers (o fins i tot els mateixos primers) pot tenir una sortida en el mercat (vaja, que algú compra primers). Una percepció completament equivocada.

Aquesta percepció es deu, possiblement, al fet que no s'expliquen mètodes per trobar primers grans als nostres graus (de matemàtiques), no surten als llibres (no especialitzats), tot i que les eines necessàries per explicar-los són simples i, de fet, formen part del currículum de primer curs de moltes de les carreres científiques (des de Matemàtiques i Física fins a moltes enginyeries: d'Informàtica, Telecomunicacions, etc).

Aquest escrit, que m'he atrevit a qualificar de manual, pretén omplir aquest buit. No explicaré res de nou ni res que no es pugui trobar en els llibres, tot i que potser sovint no es posa l'èmfasi en la practicitat que jo sí que hi posaré.

Una cosa vull dir: com que no he treballat per a la indústria dels primers, no puc saber amb seguretat si utilitza els mètodes que jo mencionaré, tot i que tinc força pistes que sí que ho fa. Sí que sé que fa servir els programes més utilitzats pels matemàtics, concretament pels aritmètics, com el SAGE, el PARI, el MAGMA, el MAPLE i altres. D'altra banda, podria haver-hi altres mètodes no públics (o secrets) que fossin millors, però d'això no en sé res (i, com s'acostuma a dir, si en sabés, ho negaria).

2. Eines preindustrials per detectar nombres primers

Tots els matemàtics sabem de l'estranya fascinació que provoquen els nombres primers. És un tema que des dels elements d'Euclides [3], on es demostrava que hi havia primers tan grans com es volgués, i possiblement molt abans, ha ocupat la ment de molts dels millors matemàtics.

Recordem que els nombres primers són els nombres que no es poden descompondre com a producte de nombres més petits (és a dir, són irreductibles) i, per tant, són els blocs a partir dels quals podem anar obtenint tots els altres nombres naturals a base de multiplicacions. La unicitat de la descomposició de tot nombre natural com a producte de primers és un dels pilars en què es fonamenta l'aritmètica (el teorema fonamental) i és el que permet demostrar el teorema de la infinitud de nombres primers. És a dir, no ens hem de preocupar per si s'acaben els primers del magatzem, perquè n'hi ha tants com en puguem necessitar.

Tot i això, hi ha moltes més preguntes pràctiques que ens podem fer sobre els primers i que no queden resoltes dient que n'hi ha infinits. En vull destacar dues: què podem fer per trobar primers molt grans i quants n'hi ha realment.

Sovint expliquem que una bona resposta a la primera pregunta és la que va donar Eratòstenes amb el seu famós mètode del garbell cap al 230 aC. La idea fonamental del seu mètode era que si volem veure que un nombre és primer, no ens cal veure que no és divisible per cap nombre més petit que ell, sinó que ho és només pels nombres més petits que la seva arrel quadrada.

La realitat, però, és que aquest mètode no serveix per a res si el nostre nombre té, per exemple, més de 50 xifres decimals. De fet, s'ha determinat que el nombre de càlculs necessaris per comprovar que un nombre de 50 xifres és primer (si ho és) amb el garbell d'Eratòstenes és molt més gran que tots els càlculs fets en tota la història de la humanitat (podeu consultar, per exemple, la pàgina 7 de [2]).

Observem que el mètode d'Eratòstenes no només prova si un nombre és o no primer, sinó que li troba un factor en cas que no ho sigui, una informació que en principi no volíem per a res si només ens interessava saber si el nombre és primer o no. Veurem que, de fet, el problema de la factorització és d'un ordre molt superior al problema de la primeritat. Així doncs, ens cal un mètode que detecti si un nombre és primer sense factoritzar-lo! La idea clau passa, curiosament, per adonar-se que és fàcil detectar nombres compostos.

3. Com detectar nombres compostos

Com podem saber que un nombre és compost sense trobar-li un factor? Sols hem de veure que **no** compleix una propietat que tinguin tots els nombres primers, encara que algun nombre compost la tingui! Una propietat, és clar, que no faci referència (directa) a la seva reductibilitat.

De propietats d'aquestes en coneixem algunes, però probablement la primera, històricament, és una de les més bones. I va ser enunciativa per qui es pot considerar el pare (modern) de l'aritmètica: Pierre de Fermat.

El 18 d'octubre de 1640 (i mentre a Catalunya començava la Guerra dels Segadors), en una carta al seu amic Frénicle de Bessy, Fermat afirmava el següent: per a tot nombre primer p , i per a tot nombre enter a , el nombre p divideix el nombre $a^p - a$. En la carta, tal com va fer en altres ocasions, mencionava que tenia una demostració d'aquest fet, però deia que no l'escribia per tal que la carta no fos massa llarga.

De fet, Fermat mateix va anunciar el seu resultat d'una manera que serà la que utilitzarem: si a no es divisible per p , aleshores p divideix $a^{p-1} - 1$.

Per tant, una manera de detectar que un nombre n és compost pot ser: escollir un nombre a no divisible per n , calcular el nombre $a^{n-1} - 1$ i veure si surt o no divisible per n . Com a cas particular tenim el criteri simple següent.

Criteri 1. *Si un nombre senar n no divideix $2^{n-1} - 1$, aleshores és compost.*

Per exemple, per al nombre $n = 143$, quan dividim $2^{142} - 1$ entre 143 ens surt un residu de 84; per tant n no és primer (de fet, $n = 11 \cdot 13$).

Compte, però, que això no serveix per detectar primers, ja que, per exemple, per a $n = 341$ tenim que $2^{n-1} - 1$ sí que és divisible per 341, però $341 = 11 \cdot 31$.

Hi ha qui ha suggerit que en casos de persones amb síndrome de Savant, que són capaces de reconèixer nombres primers força grans, de fet el que podrien estar fent és aplicar el criteri però a l'inrevés: si el nombre n divideix a $2^{n-1} - 1$, aleshores el declaren primer. Veurem que, tot i que això no és correcte, funciona prou bé en general!

Per veure que el mètode és útil, però, ens cal trobar un algorisme ràpid i eficient per decidir si $a^{n-1} - 1$ és o no divisible per n , i en particular un en què no ens calgui calcular efectivament a^{n-1} , un nombre que pot tenir un nombre de xifres impracticable.

4. El càlcul de residus, teoria i pràctica

Per tal de veure si un nombre molt gran és o no divisible per un altre nombre n no tan gran, s'utilitza el càlcul de residus, altrament dit teoria de congruències o càlcul mòdul n . El punt clau és que el residu de dividir per n la suma o el producte de dos nombres sigui igual al residu de la suma o el producte dels dos residus. Per tant, podem calcular-ho tot mòdul n .

Podem o bé utilitzar la notació deguda a K. F. Gauss, diu que $a \equiv b \pmod{n}$ per dir que a i b tenen el mateix residu en dividir-los per n , o bé considerar el conjunt $\mathbb{Z}/n\mathbb{Z} = \{0, 1, \dots, n-1\}$ de residus de manera que podem assignar a cada nombre enter un únic element de $\mathbb{Z}/n\mathbb{Z}$, i, a més, podem sumar i multiplicar aquests residus prenent el residu de l'operació com a nombre enter. Obtenim aleshores un conjunt amb dues operacions que té estructura d'anell commutatiu i, per tant, compleix (la majoria de) les propietats usuals de la suma i la multiplicació.

Compte, però, que en general no es compleix una propietat usual dels nombres: pot ser que el producte de dos nombres diferents de zero ens doni zero (habitualment es diuen divisors de zero). Per exemple, si $n = 6$, aleshores els nombres 2 i 3, mirats com a residus mòdul 6, no són zero, però sí que ho és el residu del seu producte: $6 = 2 \cdot 3$. Únicament per als nombres primers p es compleix que $\mathbb{Z}/p\mathbb{Z}$ no té divisors de zero i, de fet, és un cos, i per tant tot residu mòdul p diferent de zero té invers mòdul p . Dit d'una altra manera, per a tot nombre enter a no divisible per p , hi ha un nombre b tal que $a \cdot b - 1$ és divisible per p .

Aquest fet és la clau per als resultats bàsics que necessitem i que resumim en el resultat següent, que és una ampliació d'un resultat de Fermat.

Petit i ampliat teorema de Fermat. *Considerem p un nombre primer i denotem per $(\mathbb{Z}/p\mathbb{Z})^* := \mathbb{Z}/p\mathbb{Z} \setminus \{0\}$ amb l'operació de multiplicació de residus. Aleshores $(\mathbb{Z}/p\mathbb{Z})^*$ és un grup cíclic amb $p - 1$ elements i, per tant, existeix $g \in (\mathbb{Z}/p\mathbb{Z})^*$, de manera que per a tot element $a \in (\mathbb{Z}/p\mathbb{Z})^*$ hi ha un nombre $1 \leq i \leq p - 1$ tal que $a = g^i$. A més, els únics residus mòdul p tal que els seus quadrats són 1 són ± 1 .*

La demostració d'aquest resultat passa per uns quants resultats generals. Primer, i gràcies al teorema de Lagrange,¹ en un grup finit de d elements tots els elements compleixen que $a^d = 1$. Per tant, tots els elements del grup $(\mathbb{Z}/p\mathbb{Z})^*$ són arrel del polinomi $p(x) := x^{p-1} - 1$. Aquest polinomi no té arrels múltiples, ja que el màxim comú divisor amb la seva derivada és 1 a $\mathbb{Z}/p\mathbb{Z}$, que és un cos. Així doncs, els elements de $(\mathbb{Z}/p\mathbb{Z})^*$ són arrels $(p - 1)$ -èsimes de la unitat, i se sap que per a tot $n \geq 2$ hi ha alguna arrel primitiva enèsima de 1, de manera que totes les altres són potències d'aquesta. L'última afirmació de l'enunciat és clara, ja que en un cos els polinomis de grau d tenen com a molt d arrels.

És un problema força difícil trobar un generador del grup $(\mathbb{Z}/p\mathbb{Z})^*$ quan p és un nombre primer molt gran, però de fet només farem servir (en alguna demostració) que té algun generador.

Exemple 2. Si el nombre n no és primer, hi ha enters a tals que $a^2 \equiv 1 \pmod{n}$, però $a \not\equiv \pm 1 \pmod{n}$. Per exemple, si $n = 15$ i $a = 4$, tenim que $4^2 = 16 \equiv 1 \pmod{5}$. De fet, és fàcil de provar que si n és divisible per dos primers senars diferents, aleshores sempre hi ha un tal a .

Tornem ara al problema que ens plantejàvem al final de la darrera secció: podem determinar ràpidament si $a^{n-1} - 1$ és o no divisible per n ? Veurem que la resposta és que sí i que,

1. Val a dir que, com acostuma a passar en matemàtiques, el teorema de Lagrange no el va enunciar ell. Gauss va demostrar el cas que ens interessa. I va ser C. Jordan qui el va demostrar del tot.

de fet, hi ha un mètode immillorable per fer-ho. Abans, però, he de parlar un moment de quins algorismes poden considerar-se ràpids quan estem parlant de nombres molt grans. Direm que un algorisme que depèn d'un nombre n és lineal en n si el nombre d'operacions elementals (essencialment, suma i producte) està fitat linealment pel nombre de xifres xif(n) de n (que, essencialment, és com el logaritme $\log(n)$ de n), o sigui si hi ha nombres A i B tals que el nombre d'operacions sigui menor o igual que $A \text{ xif}(n) + B$.

Observeu que el nombre de xifres de n depèn de la base en la qual es calcula, però si ho calculem en una altra base no varia el seu caràcter (lineal, polinomial, exponencial, etcètera). La relació entre el nombre de xifres en base a i aquest nombre en base b ve donada aproximadament per la multiplicació pel $\log_a(b)$; per exemple, si un nombre té r xifres binàries, té aproximadament $\log_2(10)r \approx 0,3r$ xifres decimals. Utilitzarem xif(n) per denotar el nombre de xifres binàries.

Més en general, direm que un algorisme és polinomial si el nombre d'operacions està fitat per un polinomi en xif(n) i, en canvi, és exponencial² si està fitat per un polinomi en n i, per tant, per una funció exponencial en les xifres xif(n) de n .

El punt clau és que els algorismes lineals o polinomials augmenten la seva duració de manera controlada en augmentar les xifres de n i, en canvi, els exponencials es tornen impracticables. Per exemple, un algorisme lineal sols doblarà el temps quan doblem el nombre de xifres.

Donarem un algorisme que permet calcular potències de nombres en temps polinomial respecte a les xifres binàries xif(n) de n (de fet, el nombre de productes necessaris és lineal en xif(n)). La idea clau de l'algorisme és molt simple: si n és parell, aleshores $a^n = (a^2)^{n/2}$; i si n és senar, aleshores $a^n = a(a^2)^{(n-1)/2}$. Per tant, com a molt en dues multiplicacions reduïm el nombre de xifres binàries en 1.

Exemple 3. Imaginem que volem calcular 2^{11} . Aplicant repetidament l'observació anterior tenim

$$2^{11} = 2(2^2)^5 = 2 \left(2 \left((2^2)^2 \right)^2 \right),$$

que ens redueix a la meitat els productes necessaris per fer la potència de la manera naïf.

Si ho expressem com una funció recursiva $\text{expbin}(a, n)$, el que tenim és que

$$\text{expbin}(a, n) := \begin{cases} 1, & \text{si } n = 0, \\ \text{expbin}(a^2, \frac{n}{2}), & \text{si } n \text{ és parell } > 0, \\ a \text{ expbin}(a^2, \frac{n-1}{2}), & \text{si } n \text{ és senar.} \end{cases}$$

Si preferim l'algorisme en forma iterativa, el podem escriure de la manera següent.

2. La denominació és molt confusa quan parlem des del punt de vista del nombre n i, en canvi, és molt més clara quan parlem des del punt de vista de les xifres xif(n) de n .

Algorisme d'exponenciació binària

- (1) *Resultat* = 1
- (2) Mentre $n > 1$
- (3) Si n és senar
- (4) *Resultat* = $a * \text{Resultat}$
- (5) $n = n - 1$
- (6) $a = a^2$
- (7) $n = n/2$
- (8) *Resultat*

En el nostre cas i com que el que ens interessa és trobar la potència de a però mòdul un nombre m , els passos 4 i 6 s'han d'interpretar com que hem de fer el següent: primer calculem el producte indicat i després calculem el residu mòdul m . El càlcul d'aquest residu també és molt ràpid: lineal en el nombre de xifres binàries del producte donat, que seran com a molt el doble de les xifres de m .

En resum, l'algorisme per calcular el residu de dividir a^n entre m utilitza $2 \text{xif}(n)$ operacions «multiplicació seguida de càlcul del residu mòdul m » de dos nombres $\leq m$. Aquestes operacions necessiten, de fet, unes $2 \text{xif}(m)^4$ operacions entre bits: $\text{xif}(m)^2$ de multiplicar dos nombres de $\text{xif}(m)$ xifres, $2 \text{xif}(m)^2$ de dividir un nombre de $2 \text{xif}(m)$ xifres entre un de $\text{xif}(m)$ xifres. Per tant, si n i m tenen més o menys les mateixes xifres, com és el cas en calcular $a^{n-1} \pmod{n}$, ens calen un màxim de $4 \text{xif}(n)^5$ operacions bit per fer-ho.

5. Els primers i els secrets

Abans de veure com podem trobar nombres primers, per què ens són tan útils actualment els nombres primers. Tot passa per un mètode conegut per les inicials dels seus creadors: l'RSA (de Rivest, Shamir i Adleman). El mètode RSA és potser el mètode d'encryptació més utilitzat actualment en el món i és un mètode de clau pública que permet la comunicació secreta entre dues persones (a la pràctica, a través d'ordinadors) sense haver-se de posar d'acord prèviament en quina clau s'ha d'utilitzar. La idea és fer servir una operació que sigui molt ràpida de fer i molt lenta de desfer; en el nostre cas, la multiplicació d'enters (ràpida) i la factorització (lenta).

Podem fer una analogia física per entendre el funcionament dels mètodes d'encryptació de clau pública. Imaginem que volem rebre paquets de manera que ningú els pugui obrir abans de nosaltres. El que podem fer és repartir arreu candaus oberts (la clau pública) dels quals només nosaltres tenim la clau (la clau secreta). Qui ens vulgui enviar un paquet, l'ha d'embolicar de forma prou ferma i tancar amb un dels nostres candaus (aquesta és l'operació fàcil de fer). A partir d'aleshores només nosaltres, que tenim la clau, podem obrir el candau, i encara que algú intercepti el paquet no podrà extreure el que té al seu interior (llevat que trenqui el candau, per la qual cosa ens hem d'assegurar que és prou fort per resistir aquests atacs).

Per poder explicar el funcionament del mètode ens cal prèviament un resultat que generalitzi el petit teorema de Fermat a nombres compostos.

Petit teorema d'Euler. *Sigui $n \geq 2$ un nombre enter, i sigui $\varphi(n)$ la quantitat de nombres enters $1 \leq m \leq n - 1$ que són primers amb n . Aleshores, per a tot a enter primer amb n ,*

$$a^{\varphi(n)} \equiv 1 \pmod{n}.$$

En particular, si $n = p \cdot q$, on p i q són dos primers diferents, aleshores $\varphi(n) = n - p - q + 1$.

La demostració del resultat passa per veure que el grup dels elements invertibles respecte de la multiplicació de les classes mòdul n té $\varphi(n)$ elements (ja que la condició de ser invertible mòdul n és equivalent a la de ser primer amb n) i aplicar el teorema de Lagrange com hem fet abans per al cas n primer.

El mètode de clau pública RSA funciona de la manera següent: el receptor que vol rebre missatges té una clau secreta que està formada per la parella de primers p i q , juntament amb un nombre $2 \leq e \leq \varphi(n)$ (i primer amb $\varphi(n)$).³ Amb això podem calcular d de manera que $e \cdot d \equiv 1 \pmod{\varphi(n)}$, i de fet hi ha un algorisme en temps polinomial per fer-ho (utilitzant el conegut algorisme d'Euclides per trobar el màxim comú divisor entre dos nombres i la identitat de Bezout).

La clau pública és n i e .

Si volem enviar un missatge, per exemple, que és un nombre $2 \leq m \leq n$, calculem

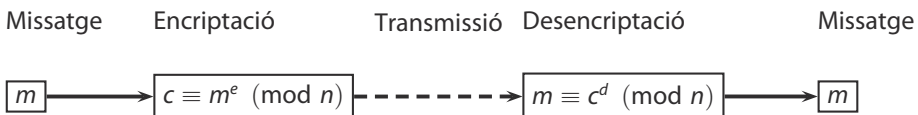
$$c \equiv m^e \pmod{n},$$

que és el que enviem.

El receptor rep c i calcula $c^d \pmod{n}$. El punt clau és que $c^d \equiv m \pmod{n}$, gràcies al petit teorema d'Euler.

Efectivament, tenim que

$$c^d = m^{ed} = m^{1+k\varphi(n)} = m \cdot (m^{\varphi(n)})^k \equiv m \cdot 1^k = m \pmod{n}$$



Si algú intercepta el missatge m i vol desencriptar-lo (o sigui, coneixent m , n i e , obtenir c), el que podria fer en teoria és el següent:

3. Sovint s'utilitza el nombre $e = 65537$ per motius de rapidesa en la computació i de seguretat. Us repto a descobrir què té d'especial aquest nombre.

1. Factoritzar n i calcular $\varphi(n)$.
2. Calcular d tal que $d \cdot e \equiv 1 \pmod{\varphi(n)}$.
3. Calcular $c^d \pmod{n}$.

Dels tres passos, només el pas 1 és difícil (en el sentit que pot dur molt de temps, com veurem més endavant). Així doncs, la seguretat del sistema depèn molt d'escollir nombres n difícils de factoritzar. En aquests moments es recomana utilitzar nombres que siguin producte de dos primers d'unes 600 xifres decimals com a mínim: el que correspon a 1024 bits, o sigui 617 xifres decimals aproximadament, i per tant claus n d'uns 2048 bits, o 1234 xifres decimals. Observeu també que cada possible receptor vol tenir un nombre n diferent, i no n'hi ha prou amb tenir uns quants primers grans, ja que provant aquests podríem factoritzar les claus. De fet, idealment, ens calen nombres primers grans escollits a l'atzar!

6. Primers industrials

Tornem ara a la nostra pregunta inicial: si volem un (i no un, sinó molts!) nombre primer d'unes 600 xifres, què podem fer per trobar-lo? La idea és simple: escollim a l'atzar un nombre senar n de les xifres desitjades i calculem $2^{n-1} \pmod{n}$ utilitzant el mètode d'exponenciació binària. Si ens surt 1, hi ha raons heurístiques per creure que aleshores el nombre és primer amb força probabilitat. Numèricament, podem veure que hi ha 78498 primers menors d'un milió i hi ha «només» 245 nombres menors d'un milió que compleixen que $2^{n-1} \equiv 1 \pmod{n}$ i no són primers.⁴ Si anem molt més enllà, hi ha 455052511 primers menors de 10^{10} , i dels altres sols 14884.⁵

Direm que un nombre n és un **primer probable**⁶ en base $a > 1$ si compleix que $a^{n-1} \equiv 1 \pmod{n}$. La idea del mètode industrial per trobar primers és buscar nombres a l'atzar que siguin pseudoprimers respecte a una base també a l'atzar. Com que és molt fàcil escollir sempre nombres senars, per exemple escollint un nombre m i després prenent $n = 2m + 1$, i ens estalviem de comprovar la meitat dels casos, és el que farem. De fet, usualment es comprova que el nombre escollit no sigui divisible per cap primer menor de 100 (o 1000), però només és una millora de rapidesa no gaire important pel que ens interessa.

Algorisme per a trobar primers industrials

Si volem un nombre primer n de N xifres decimals:

- (1) Escollim un nombre n (senar) a l'atzar amb N xifres decimals.
- (2) Escollim a l'atzar un nombre a (fins a 10000, per exemple).
- (3) Si a^{n-1} no és $\equiv 1 \pmod{n}$, tornem a 1.
- (4) Si ho és, n és un primer **industrial**.

4. S'anomenen nombres de Poulet.

5. Podeu consultar aquestes dades a A055550 del [5].

6. Noteu que els primers són primers probables, però no tots els primers probables són primers. Així doncs, estrictament parlant el nom és incorrecte: és el que en matemàtiques s'anomena un *red herring*.

Val a dir que habitualment es demana al nombre n , per tal de ser declarat primer industrial, que sigui primer probable en base a per a un nombre determinat de bases a escollides a l'atzar, habitualment 50.

És difícil determinar amb exactitud quina és la probabilitat que un primer industrial de 617 xifres de fet no sigui un primer. Però tot fa pensar que és baixíssima, i ben segur que molt més baixa que la que mai es trobarà qualsevol empresa a l'hora de fer previsions sobre el seu futur. Així doncs, des del punt de vista del negoci, el problema s'ha acabat (a part de saber aproximadament quants càlculs ens caldrà fer per trobar-ne un a la pràctica).

Però des del punt de vista d'un matemàtic, podem fer una lleugera millora al mètode que, com veurem després, ens proporcionarà una seguretat molt més gran amb molt poc esforç.

Abans d'explicar-la, però, parlarem d'uns nombres molt especials que són primers probables en totes les bases (en totes les bases que són primeres amb ell mateix) sense ser primers: els nombres de Carmichael. De fet, es poden descriure com els nombres enters n tals que tots els seus divisors primers p compleixen que $p - 1$ divideix $n - 1$. Aquests nombres es coneixen des de fa força temps i els més petits, trobats per Carmichael el 1910, són els següents:

$$561 = 3 \cdot 11 \cdot 17$$

$$1105 = 5 \cdot 13 \cdot 17$$

$$1729 = 7 \cdot 13 \cdot 19$$

$$2465 = 5 \cdot 17 \cdot 29$$

$$2821 = 7 \cdot 13 \cdot 31$$

$$6601 = 7 \cdot 23 \cdot 41$$

$$8911 = 7 \cdot 19 \cdot 67$$

Ell mateix va conjecturar que n'hi ha infinits, cosa que van poder provar el 1994 Alford, Granville i Pomerance. De fet, se sap que fins a X n'hi ha de l'ordre $> X^{2/7}$ si X és prou gran [1].

Aquests nombres de Carmichael són nombres que serien declarats primers industrials sense ser primers encara que els passéssim el test per a moltes bases. Així doncs, si podem millorar l'algorisme per evitar que hi hagi nombres que el puguin passar en moltes bases sense ser-ho, ens pot ser molt útil. I el fet és que hi ha una millora que no comporta fer més càlculs, ja que aprofita els càlculs que calen per mirar si un nombre és un primer probable.

La idea per millorar el mètode és utilitzar el que ja he dit en el petit i ampliat teorema de Fermat: un residu r a $\mathbb{Z}/p\mathbb{Z}$ per a p primer tal que $r^2 \equiv 1$, ha de complir que $r \equiv \pm 1$. Per tant, si p és primer senar i a és un nombre enter no divisible per p , tenim que:

$$a^{\frac{p-1}{2}} \equiv \pm 1 \pmod{p}.$$

Com que podria ser que $p - 1$ fos divisible per 2 més d'una vegada, podem utilitzar aquest criteri repetidament.

Criteri 4. Donat n un enter senar, escrivim $n - 1 = 2^s \cdot r$ amb r senar. Si a és un enter primer amb n, i

$$(1) a^r \not\equiv 1 \pmod{n}$$

$$(2) i$$

$$a^{r \cdot 2^t} \not\equiv -1 \pmod{n}$$

per a tot $0 \leq t \leq s - 1$,

aleshores n és compost.

Un nombre enter n tal que en aplicar aquest criteri per a donat no doni compost, es diu que és un **primer fortament probable** en base a .

Observeu que el criteri funciona, ja que si el nombre n és primer, aleshores $a^{n-1} \equiv 1 \pmod{n}$ i, per tant, si $a^r \not\equiv 1 \pmod{n}$ i com que $(a^r)^{2^s} \equiv 1 \pmod{n}$, per a alguna t tenim que $(a^r)^{2^t} \pmod{n}$ és un nombre que no és 1 però el seu quadrat ho és, i per tant ha de ser -1 .

Fixeu-vos que els càlculs necessaris no augmenten gens respecte als dels primers probables, ja que acabem calculant les potències de a que ja hauríem calculat per veure que és primer probable en base a .

Exemple 5. Considerem el nombre $n = 561$, que és el menor nombre de Carmichael. Tenim que $n - 1 = 2^4 \cdot 35$. Calculem:

- $2^{35} \equiv 263 \pmod{561}$
- $2^{2 \cdot 35} \equiv 166 \pmod{561}$
- $2^{2^2 \cdot 35} \equiv 67 \pmod{561}$
- $2^{2^3 \cdot 35} \equiv 1 \pmod{561}$

i, per tant, és compost seguint el criteri 4.

Utilitzant aquest criteri obtenim un algorisme lleugerament millorat per trobar primers industrials.

Algorisme millorat per trobar nombres primers industrials

Si volem un nombre primer n de N xifres decimals:

- (1) Escollim un nombre n (senar) a l'atzar amb N xifres decimals.
- (2) Escollim a l'atzar un nombre a (fins a 1000, per exemple).
- (3) Si n no és un pseudoprimers fort en base a , anem a (1).
- (4) Si ho és, n és un primer industrial fort.

Com en el cas dels primers industrials, normalment es demana que sigui un primer fortament probable per unes quantes bases, habitualment 50.

Veurem més endavant que si un nombre és un primer fortament probable per a un nombre prou gran de bases, però no massa gran (de fet, el nombre de bases necessàries depèn linealment de les xifres del nombre en qüestió), podem estar del tot segurs (almenys si ens creiem una conjectura molt coneguda anomenada hipòtesi de Riemann generalitzada) que el nombre serà primer. Així doncs, des del punt de vista teòric i pràctic, aquest mètode és imbatible.

7. Un criteri a la Fibonacci

Comentarem un criteri diferent, d'alguna manera ortogonal, per veure que un nombre és probablement primer. La idea, com abans, és veure si es compleix una propietat que és certa per als primers i que és fàcilment calculable. En aquesta secció haurem d'utilitzar alguns resultats més avançats de matemàtiques per a les demostracions.

Considerem les successions del tipus Fibonacci donades de la forma següent. Fixem a , un nombre enter, i considerem les successions $u_n(a)$ i $v_n(a)$, donades les dues per la mateixa recurrència

$$u_n(a) = u_{n-1}(a) - au_{n-2}(a) \text{ si } n \geq 2$$

però amb termes inicials diferents: $u_0(a) = 0$ i $u_1(a) = 1$, i $v_0(a) = 2$ i $v_1(a) = 1$. Fixeu-vos que la successió de Fibonacci és $u_n(-1)$.

De la mateixa manera que es pot veure que la successió de Fibonacci es pot expressar en termes del nombre d'or $\frac{1+\sqrt{5}}{2}$, aquestes successions es poden posar en funció de les arrels $\alpha(a) := \frac{1+\sqrt{1-4a}}{2}$ i $\beta(a) := \frac{1-\sqrt{1-4a}}{2}$ (reals o complexes) del polinomi $x^2 - x + a$.

De fet, tenim que

$$v_n(a) = \alpha(a)^n + \beta(a)^n \quad \text{i} \quad u_n(a) = \frac{\alpha(a)^n - \beta(a)^n}{\sqrt{1-4a}},$$

que ens permet veure les fórmules recursives següents:

$$\begin{aligned} u_{2n}(a) &= u_n(a)v_n(a) & u_{2n+1}(a) &= \frac{u_{2n}(a) + v_{2n}(a)}{2} \\ v_{2n}(a) &= v_n^2 - 2a^n & v_{2n+1} &= \frac{(1-4a)u_{2n}(a) + v_{2n}(a)}{2} \end{aligned}$$

El següent lema és l'equivalent en el món de les successions del petit teorema de Fermat.

Lema de Lucas. *sigui p nombre primer de manera que $1 - 4a$ no sigui congruent mòdul p en cap nombre enter al quadrat. Aleshores*

$$u_{p+1}(a) \equiv 0 \pmod{p}.$$

La demostració s'utilitza que si $1 - 4a$ no és un quadrat mòdul p , aleshores el polinomi $x^2 - x + a$ no té arrels mòdul p . Per tant, les arrels $\alpha(a)$ i $\beta(a)$ mòdul p viuen en un cos més gran, que necessàriament ha de ser el cos de p^2 elements \mathbb{F}_{p^2} .

En aquest cos es té en general que si α i β són les dues arrels d'un polinomi $f(x)$ quadràtic i irreductible amb coeficients a \mathbb{F}_p , aleshores $\alpha^p = \beta$. Això és degut que «elevant a p » respecta les operacions de suma i producte a \mathbb{F}_{p^2} (i a qualsevol cos que contingui \mathbb{F}_p), d'on es dedueix que α^p ha de ser una arrel de $f(x)$, i com que no pot ser α , ha de ser l'altra arrel β .

Obtenim, per tant, que $\alpha(a)^{p+1} \equiv \beta(a)^{p+1}$ mòdul p . Utilitzant les fórmules anteriors obtenim el resultat.

El problema per poder utilitzar aquest resultat passa per saber decidir si el meu nombre $1 - 4a$ és o no és congruent amb algun nombre enter al quadrat. La idea és utilitzar la famosíssima llei de reciprocitat quadràtica de Gauss,⁷ de la qual es dedueix el resultat següent:

Teorema 6. Prenem $d = 1 - 4a$ (que és un enter senar (positiu o negatiu) i $d \equiv 1 \pmod{4}$) i p un nombre primer senar. Suposem que d és lliure de quadrats. Aleshores d és un quadrat mòdul p si p és un quadrat mòdul $|d|$.

El resultat, de fet, és un si i només si quan $|d|$ és un nombre primer.

Obtenim així el criteri següent.

Criteri 7. Sigui n un nombre senar. Busquem el menor enter d de la llista $\{5, -7, -11, 13, -15, \dots\}$ dels enters senars lliures de quadrats (amb el signe adequat per tal que $d \equiv 1 \pmod{4}$) de manera que n no és un quadrat mòdul $|d|$. Sigui $a = \frac{1-d}{4}$.

Si $u_n(a) \not\equiv 0 \pmod{n}$, aleshores és compost.

Observeu que per calcular $u_n(a)$ utilitzant les fórmules recursives anteriors, ho podem fer d'una manera similar a l'exponenciació binària. Deixo al lector com a exercici l'algorisme en qüestió que el calcula.

Val a dir que hi ha un test de Lucas fort, en un sentit similar al criteri 4 dels primers fortament probables en base a .

Criteri 8. Sigui n un nombre senar. Busquem el menor enter d de la llista $\{5, -7, -11, 13, -15, \dots\}$ dels enters senars lliures de quadrats (amb el signe adequat per tal que $d \equiv 1 \pmod{4}$) de manera que n no és un quadrat mòdul $|d|$. Sigui $a = \frac{1-d}{4}$.

Escrivim $n + 1 = 2^s \cdot r$, amb r senar.

Si $u_r(a) \not\equiv 0 \pmod{n}$ i $v_{r2^t}(a) \not\equiv 0 \pmod{n}$ per a tot $0 \leq t < s$, aleshores n és compost.

7. Enunciar i demostrar aquesta llei ens portaria per camins molt interessants, però potser allunyats de l'objectiu d'aquest manual. D'altra banda, recomano al lector interessat explorar a [4] alguna de les 246 demostracions conegudes d'aquesta llei, començant per les sis que va escriure Gauss.

Un nombre que superi aquest test (o sigui, que no digui que és compost) s'anomena primer fortament probable de Lucas.

Combinant els tests que hem presentat obtindrem el que s'anomena el test de primeritat de Baillie-PSW, que és el que alguns programes d'ordinador apliquen quan els demanem si un nombre prou gran és primer.

Algorisme més utilitzat per trobar nombres primers industrials

Si volem un nombre primer n de N xifres decimals:

- (1) Escollim un nombre n (senar) a l'atzar amb N xifres decimals.
- (2) Mirem si és primer fortament probable en base 2.
- (3) Mirem si és primer fortament probable de Lucas.
- (4) Si supera els dos tests, declarem n primer industrial de Baillie-PSW.

Val a dir que no es coneix **cap** nombre senar n que sigui primer industrial de Baillie-PSW i no sigui primer.⁸ De fet, s'ha comprovat que tots els nombres amb menys de 64 bits que passen aquest test són primers. Tot i això, hi ha arguments heurístics per justificar que hi ha d'haver infinits nombres que superin aquest test i no siguin primers.

8. Quina és la probabilitat que un nombre sigui primer?

Per tal que els mètodes que he explicat en la secció anterior siguin realment efectius, cal saber quants primers hi ha d'un nombre donat de xifres; o, dit d'una altra manera, quina és la probabilitat (aproximada) que un nombre de n xifres sigui primer.

La resposta a aquesta pregunta, encara que ens pugui semblar mentida, és coneguda des de fa molt temps. De fet, el mateix K. F. Gauss als setze anys (el 1791) va conjecturar que el nombre de primers de n xifres hauria de ser aproximadament de $n/\log(n)$. Sembla que va fer aquesta hipòtesi després de comptar nombres de primers de mil en mil, que permet eliminar les fluctuacions en el nombre quan es consideren intervals massa petits (tal com si fossin errors experimentals). Altres matemàtics van fer i publicar conjectures equivalents, fins que l'any 1896 Jacques Hadamard i Charles Jean de la Vallée-Poussin van demostrar el resultat simultàniament i de manera independent, basant-se en les idees introduïdes per P. L. Txeixov i G. F. B. Riemann.

Com a conseqüència, podem calcular quants nombres a l'atzar haurem d'escollir en els algorismes per obtenir primers industrials explicats anteriorment per tal de trobar un primer. Com que $\log(10^{617}) \approx 1421$, la probabilitat que un nombre de 617 xifres sigui primer és aproximadament d'una entre 1421; o sigui, entre 710 si només agafem senars. Així doncs, de mitjana haurem d'escollir uns 700 nombres senars de 617 xifres decimals a l'atzar per trobar-ne algun que sigui nombre primer.

8. Fins i tot hi ha un premi en metàl·lic per a qui en trobi un (de només 30 dòlars!), ofert per Pomerance, Selfridge i Wagstaff. Un exemple anàleg però per a un criteri similar amb els nombres de Fibonacci té una recompensa més alta (de 635 dòlars): s'anomena la conjectura de Selfridge.

Fixeu-vos que en doblar les xifres la probabilitat «sols» es divideix per 2. Per tant, si volem un primer amb el doble de xifres, 1234 xifres, haurem de repetir l'algorisme només el doble de cops.⁹

9. Primers computacionals: mètodes i certificats

Imaginem ara que tenim un nombre del qual volem demostrar matemàticament que és un nombre primer (i no solament estar-ne molt segurs). Podem fer-ho utilitzant les idees explicades anteriorment? Resulta que sí que podem, gràcies a un resultat degut al matemàtic francès E. Lucas (i millorat per uns quants autors).

Test de Lucas. Donat $n > 1$ enter, si existeix un enter $a > 1$ tal que $a^{n-1} \equiv 1 \pmod{n}$ i, per a tot primer p dividint $n - 1$,

$$a^{(n-1)/p} \not\equiv 1 \pmod{n},$$

aleshores n és primer.

La demostració del resultat és força senzilla. En efecte, si existeix un tal a , aleshores a és invertible a $\mathbb{Z}/n\mathbb{Z}$ i, a més, té ordre $n - 1$. Però el nombre d'invertibles a $\mathbb{Z}/n\mathbb{Z}$ és menor estrictament que $n - 1$ si n no és primer, d'on deduïm el resultat. Que per als nombres primers es compleix el resultat es deu a l'ampliació del petit teorema de Fermat que hem explicat a la secció 5; el nombre a que demana el teorema és qualsevol generador del grup $(\mathbb{Z}/n\mathbb{Z})^*$

De fet, aquest test es pot millorar de diverses maneres. Per exemple, en lloc de trobar un enter a que valgui per a tots els divisors primers p de $n - 1$, podem trobar un a_p que ens serveixi per a cada p .

Una altra millora, aquesta més interessant, és que no ens cal veure-ho per a tots els divisors primers, sinó per a suficients primers: si ho sabem per a tots els primers dividint m de manera que $n - 1 = md$, amb $m > \sqrt[n]{n}$ ja en tenim prou (una millora que devem a Pocklington).

Fins i tot hi ha millores que permeten considerar una factorització, però aquest cop amb $m > \sqrt[3]{n}$, o fins i tot una mica més petit, però no les explicarem aquí.¹⁰

Exemple 9. Considerem $n = 6271$. Tenim que $n - 1 = 11 \cdot 19 \cdot 30$ i $30 < \sqrt{n} \approx 71$.

Denotem per $d_1 := (n - 1)/11 = 570$ i $d_2 := (n - 1)/19 = 330$.

Calculem

$$2^{d_1} = 2^{570} \equiv 4365 \pmod{6271}$$

9. Per donar una idea del temps necessari he fet personalment l'experiment d'executar un programa simple sense optimitzar per calcular primers industrials forts, i li calen de mitjana menys de 4 segons per trobar-ne algun amb 617 xifres.

10. Podeu consultar-les al capítol 4 de [2].

$$2^{d_2} = 2^{330} \equiv 5016 \pmod{6271}$$

mentre que

$$2^{n-1} \equiv 1 \pmod{6271}$$

Per tant, 6271 és primer.

De fet, podríem comprovar fàcilment que $a = 2$ compleix també les altres condicions per a la resta de primers.

Us podeu preguntar si és gaire difícil o no trobar un tal nombre a si n és primer (que s'anomena arrel primitiva mòdul n): la resposta és que no. De fet, se sap que si n és més gran que el producte dels 11 primers nombres primers, aleshores el nombre de vegades que haureu d'escollir un a a l'atzar fins a trobar-ne un de bo és $2 \log(\log(n))$, que és molt petit.¹¹ La dificultat més gran és factoritzar $n - 1$; per això és important la millora de Pocklington.

De totes maneres, fins i tot quan puguem aplicar aquest criteri no està gens clar que tinguem **realment** una demostració que un nombre donat és primer. Perquè, imagineu que després d'un mes de càlculs amb l'ordinador obteniu com a resposta un **sí** pelat a la pregunta de si un nombre concret és primer; podem considerar que això és una demostració?

Per aquest motiu, si es necessita una demostració que un nombre donat és primer, normalment es demana un certificat de primeritat. No, no es tracta d'un document on algú amb molta autoritat (jo mateix, per exemple!) certifiqui que el nombre és primer. Es tracta de donar unes dades que permetin repetir el càlcul que ens dona la primeritat, però aquest cop amb molta rapidesa.

Per al cas del test de Lucas (o per al de Pocklington), és molt fàcil de descriure. Imagineu que per a un nombre n doneu els factors primers de $n - 1$ i el nombre a del criteri: comprovar aleshores que es compleix el criteri per a aquests nombres és molt ràpid (per a un ordinador). Observeu, però, que en donar els factors primers també esteu assegurant que aquests nombres són primers i, per tant, potser haureu de donar un certificat també per a aquests nombres.

10. Com factoritzar com un campió

Ja hem vist que per poder demostrar que un nombre és primer utilitzant el test de Lucas ens cal saber factoritzar. De fet, si volem utilitzar els nostres nombres primers per construir claus de l'RSA (multiplicant dos primers grans), ens pot ser molt útil saber què cal fer per trencar aquesta clau (o sigui, factoritzant el nombre).¹²

Ens trobem en la situació següent: tenim un nombre que sabem que és compost (per exemple, perquè no és un primer probable en alguna base) i volem trobar-ne un factor no

11. Vegeu per exemple l'exercici 4.1 de [2].

12. No és cert que el problema de factoritzar sigui equivalent al de trencar l'RSA, ja que hi ha mètodes que en alguns casos poden desxifrar certs missatges xifrats amb RSA sense factoritzar.

trivial. Si el meu nombre és un nombre a l'atzar és molt probable que tingui algun factor primer petit. Ens situarem, però, en el cas que el meu nombre no tingui un factor petit; per tant, suposarem que el nostre nombre no és parell, ni múltiple de 3, etcètera. Recordem que pretendre factoritzar un nombre que no sigui producte d'un primer petit (de menys de 20 xifres, per exemple) utilitzant l'algorisme d'Eratòstenes, és una idea pèssima. Així doncs, us proposo algunes idees per factoritzar (amb ordinador) de manera molt més eficient.

10.1. Comencem per dalt

Aquest mètode vindria a ser una variant del mètode d'Eratòstenes però començant «per dalt». De manera naïf, el que podríem fer per factoritzar un nombre n és prendre $r := \lfloor \sqrt{n} \rfloor$, que és el nombre enter més proper per sota a l'arrel quadrada,¹³ i ara provar si n és divisible per r , per $r - 1$, etcètera. El mètode, però, sols funciona bé si el nombre n és producte de dos nombres molt propers. És molt millor una lleugera variant del mètode. El que farem és prendre $(r + 1)^2 - n$ i mirar si és o no un quadrat. Si ho és, tenim $a^2 = b^2 - n$ (on $a = r + 1$) i, per tant, $n = (b + a)(b - a)$ i, per tant, una factorització no trivial. Si no, augmentem r en 1 i repetim.

Exemple 10. Considerem

$$n = 14979829200673042122247562983237289188933$$

(l'he trobat fent producte de dos primers «quasiconsecutius» a l'atzar). Aleshores

$$r = 122392112493710321292.$$

Tenim que $(r + 1)^2 - n = 54^2$ i, per tant,

$$n = (r + 1 + 54)(r + 1 - 54) = 122392112493710321347 \cdot 122392112493710321239.$$

Aquest mètode s'anomena de Fermat, tot i que no tinc clar que Fermat mateix fos qui el va desenvolupar. Una cosa curiosa és que alguns dels programes que factoritzen nombres no proven per defecte amb aquest mètode abans d'aplicar-ne altres de més sofisticats, de manera que podeu vèncer un programa com aquests amb aquest mètode per a aquest tipus de nombres.

10.2. Ens ho juguem al casino

En lloc de començar per baix o per dalt, podem provar sort a veure si trobem algun factor a l'atzar. Però és clar que necessitem alguna manera d'incrementar les nostres possibilitats que no sigui purament escollir nombres a l'atzar menors que \sqrt{n} i mirar si divideixen n . La primera idea és utilitzar la (mal anomenada) paradoxa de l'aniversari: que diu que si escollim prou (però menys de les que un creuria *a priori*) persones a l'atzar, hi ha molta probabilitat que n'hi hagi dues que celebren el seu aniversari el mateix dia. De fet, si són tan sols 23, persones

13. Fàcilment calculable, per exemple, utilitzant el mètode de Newton.

la probabilitat és superior al 50%. Així, podríem escollir uns quants nombres a l'atzar entre 1 i n , i prenem totes les restes dos a dos. Amb sort, dos dels nombres seran equivalents mòdul factor de n , que podem trobar fent el màxim comú divisor amb n .

El que farem és una modificació d'aquesta idea, deguda a J. Pollard, de l'any 1975 i que s'anomena la ρ de Pollard. La idea és considerar una funció g del conjunt $\{1, \dots, n\}$ que es comporti de manera similar a una funció a l'atzar. La funció que ell va considerar era $g(x) = x^2 + 1 \pmod{n}$, però (gairebé) qualsevol polinomi de grau > 1 aniria bé. Els elements que construïm «a l'atzar» consisteixen a aplicar repetidament la funció g . A la pràctica ell proposa l'estratègia següent, basada en l'algorisme de la llebre i la tortuga de Floyd: escollim un nombre a i prenem la parella $(r, s) := (g(a), g^2(a))$ com a inicial. Si $1 < \text{mcd}(n, r - s) < n$, ja tenim un factor. Si no, calculem la nova parella $(r, s) = (g(r), g^2(s))$ i repetim.

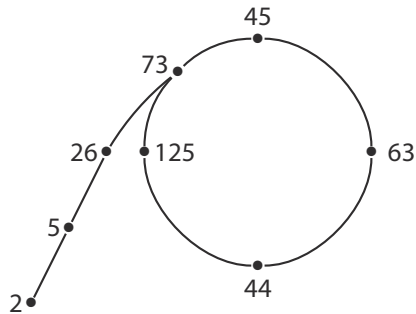
Algorisme ρ de factorització

- (1) Escollim un nombre $1 \leq a \leq n$ a l'atzar
- (2) $x = a, y = a, d = 1$
- (3) Mentre $d = 1$
- (4) $x = x^2 + 1 \pmod{n}$
- (5) $y = y^2 + 1 \pmod{n}$
- (6) $y = y^2 + 1 \pmod{n}$
- (7) $d = \text{mcd}(x - y, n)$
- (8) Si $d = n$, aleshores ves a (1)
- (9) d

Exemple 11. Considerem el nombre $n = 46357$. Escollim $a = 2$ i apliquem el mètode repetidament. Obtenim les parelles (x, y) següents: $(5, 26)$, $(26, 41117)$, $(677, 32962)$, $(41117, 39937)$, $(14257, 16220)$. Per a aquesta última parella tenim que $\text{mcd}(y - x, n) = \text{mcd}(1963, 46357) = 151$, que és un factor no trivial de n . Finalment, tenim que $n = 151 \cdot 307$.

De fet, podeu comprovar que el mòdul 151, la funció $g(x) = x^2 + 1$, començant per $x = 2$ pren els valors $2, 5, 26, 73, 45, 63, 44, 125, 73 \dots$, i es repeteixen a partir d'aquí. Si ho dibuixem, obtenim la forma de ρ característica del mètode.¹⁴

14. De fet, s'anomena mètode ρ per la forma que fa un cicle, «amb cua», com el que fa les iteracions de g .



10.3. Com més serem, millor factoritzarem!

Els millors mètodes per factoritzar actualment permeten subdividir els càlculs entre molts ordinadors. La idea prové del mètode de Fermat: per tal de factoritzar n en tenim prou amb trobar dos enters x i y tals que $x^2 \equiv y^2 \pmod{n}$ i $x \not\equiv \pm y \pmod{n}$: el factor que busquem serà $\text{mcd}(x - y, n)$. El que es fa es trobar un procediment que permet trobar parelles (x, y) tals que $x^2 \equiv y^2 \pmod{n}$, esperant trobar-ne alguna no trivial (a la pràctica la majoria ho són).

La idea és simple: en lloc de trobar dos quadrats iguals mòdul n , el que farem és trobar quadrats iguals a productes de primers petits mòdul n . Si en tenim prou, podem multiplicar-los adequadament fins a trobar un quadrat. Aquest mètode va ser desenvolupat per primer cop als anys 1920 per Maurice Kraitchik i ha estat millorat al llarg dels anys fins a arribar a l'anomenat garbell dels cossos de nombres, que és el millor mètode conegut actualment per factoritzar nombres grans.

Exemple 12. Factoritzem per aquest mètode el nombre $n = 15347$. Si prenem $r = 124$, que és la part entera per sobre de \sqrt{n} , aleshores $n - r^2 = 29$. Tenim les igualtats següents:

$$124^2 \equiv 29 \pmod{n}$$

$$127^2 \equiv 2 \cdot 17 \cdot 23 \pmod{n}$$

$$195^2 \equiv 2 \cdot 17 \cdot 23 \cdot 29 \pmod{n}$$

Per tant, el seu producte és un quadrat; o sigui, si prenem

$$a = 124 \cdot 127 \cdot 195 \equiv 1460 \pmod{n}$$

i

$$b = 2 \cdot 17 \cdot 23 \cdot 29 = 22678,$$

aleshores $b - a = 21218$ i $\text{mcd}(b - a, n) = 103$, que és un factor. Finalment, obtenim $n = 103 \cdot 149$.

Expliquem una mica més la idea bàsica. Primer cal escollir quants primers (i quins primers) considerarem com a primers petits: s'anomenen la base de factors. Cada cop que es troba un quadrat igual a un producte de primers de la base de factors, guardem el vector format per les potències en què apareixen aquests factors. El problema de trobar un producte de tals nombres que sigui un quadrat esdevé un problema de trobar una suma dels vectors que sigui múltiple de 2; i aquest problema, a més, es pot resoldre fàcilment utilitzant àlgebra lineal sobre el cos de dos elements \mathbb{F}_2 i considerant els vectors mòdul 2.

És clar que el punt clau és com podem obtenir nombres el quadrat dels quals sigui congruent amb el producte de primers petits. Però el gran avantatge del mètode és que podem tenir diversos ordinadors (milions d'ordinadors, si cal) buscant aquest tipus d'igualtats, i quan en tenim prou les ajuntem i resollem l'àlgebra lineal a \mathbb{F}_2 en un ordinador central.

Amb aquests mètodes es va aconseguir factoritzar nombres «difícils» de fins a 232 dígits decimals (768 bits) l'any 2009, en el que s'anomena el repte RSA, després de dos anys de feina de tot un equip de persones. Els programes públics poden factoritzar nombres d'uns 500 bits en un temps «raonable». Això ha causat controvèrsies com el cas de la clau de signatura de Texas Instruments, en què un usuari sense coneixements matemàtics especials va factoritzar la seva clau RSA a l'estiu de l'any 2009 en una computació que va trigar uns 73 dies i després la va fer pública, cosa que va causar una disputa sobre si podia o no fer-ho legalment (la resposta oficial li va donar la raó, finalment).

11. Els primers són P

Un cop hem vist que no coneixem algorismes ràpids (ni polinomials) per factoritzar nombres, que és el que necessitem per poder aplicar el test de Lucas a un nombre, ens podem preguntar si tot i això hi ha algun algorisme que en temps polinomial (respecte a les xifres del nombre) ens demostrï si el nombre és o no primer. I la resposta, curiosament, és que sí que en coneixem!

El primer algorisme (i potser el millor algorisme) per demostrar que un nombre és primer en temps polinomial el va calcular Miller¹⁵ el 1976. Només té una petita pega: per tal d'estar segurs de la veracitat del mètode, ens hem de creure una conjectura.¹⁶ Però no una conjectura qualsevol, no. Es tracta probablement de la conjectura més famosa de les matemàtiques: la hipòtesi de Riemann. De fet, ens cal una versió més general, anomenada la hipòtesi de Riemann generalitzada. Donem primer el mètode i després comentaré com s'utilitza aquesta conjectura.

Criteri 13 (Miller). *Si la conjectura generalitzada de Riemann és certa i un nombre n és un primer fortament probable en base a per a tot $2 \leq a \leq 2(\log(n))^2$, aleshores és primer.*

Noteu que el mètode per decidir si un nombre n és primer fortament probable en base a és polinomial en el nombre de xifres de n ; i el teorema ens diu que l'hem d'aplicar $2(\log(n))^2$ vegades. Per tant, el mètode que ens dona el teorema és en temps polinomial.

15. La versió que donem aquí és de Bach.

16. Cosa que pot canviar si algun dia algú la demostra.

I com s'utilitza aquesta conjectura per demostrar el resultat? Doncs, de fet, el que s'utilitza només és que la conjectura generalitzada de Riemann implica que per a tot nombre primer p hi ha algun nombre $a < 2(\log(p))^2$, de manera que a no és un quadrat mòdul p .

Així doncs, durant molt temps va estar oberta la qüestió de si existia un mètode per provar primeritat que fos en temps polinomial i la veracitat del qual es pogués demostrar sense apel·lar a cap conjectura com la hipòtesi de Riemann. Això va ser resolt l'agost del 2012 per M. Agrawal, N. Kayal i N. Saxena [6], que van donar un mètode (anomenat AKS, per les sigles dels seus autors) per demostrar la primeritat en temps polinomial sense usar cap conjectura. El test, de fet, és molt simple i utilitza una versió amb polinomis del test de Fermat dels primers probables. També és sorprenent que els autors no eren matemàtics teòrics, sinó de teoria de la computació, i dos d'ells (Kayal i Saxena), de fet, estaven fent la tesi de màster a Kanpur (Índia) i Agrawal era el seu director.

El mètode AKS té, però, un problema. I és que tot i ser en temps polinomial, és molt més lent que els mètodes ja coneguts pels primers «petits» del tipus que normalment ens trobem: o sigui, amb els que podem realment calcular. Encara que això us resulti sorprenent, té relació amb el fet que ser en temps polinomial és bo a la llarga, però les constants involucrades poden ser enormes i, per tant, per als nombres que ara per ara podem utilitzar és prohibitiu.

En resum, el mètode AKS és, tot i que sembla un mètode pràctic, més un resultat teòric que demostra l'existència d'un bon mètode que no pas un resultat utilitzable a la pràctica.

12. El futur de la factorització

Com que les meves dots premonitòries són més aviat escasses, el que diré en aquesta última secció ho haurem de deixar com una simple especulació. Però sí que voldria comentar un parell d'idees sobre què ens pot oferir el futur respecte a aquest problema. I és que, encara que us pugui semblar un tema purament acadèmic, l'obtenció d'un mètode/procediment/aparell que permeti factoritzar nombres molt grans de manera pràctica podria provocar o bé una gran sacsejada a l'economia mundial si se sabés públicament, o bé que la persona o l'organisme que ho sabés fer tindria un poder immens per la possibilitat d'interceptar les comunicacions secretes mentre no s'implementés massivament una nova tècnica d'encryptació pública (cosa que no passaria mentre no fos pública l'existència de la tècnica).¹⁷ És clar que hom podria especular si això no passa ja realment.

En primera instància, ens podem plantejar si és possible l'existència d'un algorisme que permeti factoritzar en temps polinomial. A diferència del cas ja tractat de decidir si un nombre és primer, en què prèviament al teorema AKS ja se sabia que l'existència d'un mètode en temps polinomial es podia deduir d'una conjectura prèvia força creïble, la majoria d'experts consideren que no existeix un tal mètode per a la factorització.¹⁸

El problema de la factorització està molt relacionat amb problemes bàsics de la teoria de la computació, però no és un problema central. D'una banda, no és cert que l'existència d'un

17. Podeu veure a la pel·lícula *The Travelling Salesman* una discussió de tot això en una ficció molt realista.

18. En la meua humil impressió.

tal algorisme aportés una solució al problema més important respecte a la computabilitat, l'anomenat problema P versus NP, que pregunta si és cert que tot problema una solució del qual pot ser verificada en temps polinomial (com la factorització) pot ser de fet resolt en temps polinomial. Se sap que hi ha problemes tals que, si es pogués demostrar l'existència d'un algorisme per resoldre'ls en temps polinomial (o, millor encara, trobar-lo), aleshores tindriem resolt afirmativament el problema P versus NP: s'anomenen problemes NP-complets. Alguns exemples coneguts són el problema del viatjant de comerç, el problema de la motxilla o el colorit de grafs. Però no se sap si el problema de la factorització és NP-complet, tot i que hi ha motius teòrics per creure que no ho és. D'altra banda, és clar que si es pogués demostrar que no hi ha un algorisme per factoritzar en temps polinomial, aleshores el problema P versus NP quedaria resolt negativament.

En segona instància, hi ha la possibilitat que es pugui fabricar un aparell que pugui factoritzar en temps polinomial. De fet, sorprenentment, aquest aparell ja ha estat dissenyat teòricament i fins i tot s'han construït prototips que ho fan... amb nombres molt petits. Es basa en el que s'anomena la computació quàntica, que permetria, mitjançant la superposició d'estats, detectar l'ordre r d'un enter a mòdul n en temps polinomial. Si aquest ordre és parell, aleshores $a^{r/2}$ mòdul n és un nombre el quadrat del qual és 1, i si no és ni 1 ni -1 , ens dona un factor de n calculant simplement $\text{mcd}(a^{r/2} - 1, n)$. No tinc clar fins a quin punt s'ha factoritzat nombres grans amb aquesta idea: se sap que un ordinador quàntic va factoritzar 21 utilitzant aquesta tècnica el 2012 (fet que possiblement no us deua impressionar gaire), però sembla que hi ha hagut alguna millora substancial recentment. De totes maneres i per motius obvis que ja he explicat, podria haver-hi un cert secretisme respecte al que realment se sap fer.

Així doncs, podeu pensar que en poc temps es poden produir avenços molt grans en aquest problema, tot i que he de dir que de tot això ja fa temps que en parlem, i estem essencialment igual. Així, podem esperar que aquest manual seguirà vigent d'aquí a uns quants anys. I si voleu saber més dels primers, aquest cop sobre els seus aspectes més teòrics, no us perdeu el meu propi article [7].

Agraïments

Tot i que m'he dedicat a l'aritmètica durant tota la meua carrera professional i que els nombres primers són omnipresents en la meua recerca, la veritat és que fins fa molt poc tots els primers que hi apareixien eren minúsculs o bé indeterminats (denotats per una simple lletra). Aquesta situació va canviar quan vaig començar a impartir l'assignatura d'aritmètica alguns dels darrers anys. Va ser aleshores quan em vaig preguntar com es feia **realment** per aconseguir nombres primers útils (criptogràficament parlant). He de dir que aquesta era una pregunta que em rondava pel cap des que l'Amparo López i l'Enric Nart em van parlar per primer cop de les aplicacions de l'aritmètica a la criptografia (ja fa potser massa anys), però per algun motiu no vaig arribar mai a verbalitzar-la. Així doncs, els ho agraeixo a ells dos, així com als alumnes que en els darrers anys han hagut de veure com jo anava aprenent amb ells.

Finalment, voldria agrair a W. Pitsch que em demanés de fer la conferència inaugural dels graus de Matemàtiques i Estadística que són l'origen d'aquest manual (i d'un altre *survey* [6]), al *referee* desconegut (per a mi) pels comentaris, suggeriments i correccions proposades, així com la insistència de J. Ll. Solé perquè l'escriués.

Referències

- [1] W. R. Alford, A. Granville i C. Pomerance. (1994). « There are Infinitely Many Carmichael Numbers», *Annals of Mathematics*, 139, 703-722.
- [2] R. Crandall i C. Pomerance. (2005). *Prime numbers: a computational perspective*, 2 a ed. Nova York. Springer.
- [3] Euclides. *Els elements*, circa 300.
- [4] F. Lemmermeyer. (2000). *Reciprocity Laws. From Euler to Eisenstein*. Berlín. Springer.
- [5] *The On-Line Encyclopedia of Integer Sequences*, publicada electrònicament a <https://oeis.org>.
- [6] M. Agrawal, N. Kayal, i N. Saxena. (2004). «PRIMES is in P». *Annals of Mathematics*, 160(2): 781-79.
- [7] X. Xarles. «Els primers, desvelats». Per aparèixer al *Butlletí de la Societat Catalana de Matemàtiques*.

