ugr Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

PRIMALIDAD. TESTS Y CERTIFICADOS

¿Cómo se prueba que n es primo?

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

1. ALGORITMOS DETERMINISTAS VS PROBABILÍSTICOS

Un algoritmo (determinista) es un proceso computacional que dada una entrada (input) produce una salida (output) en un número finito de pasos (**iteraciones**) donde en cada paso se introduce la entrada producida por el anterior y se genera una salida para el siguiente efectuando las mismas operaciones.

Si el número de pasos está acotado por una función polinómica que depende sólo de la entrada inicial. El problema que resuelve (pregunta que contesta) está en la **clase P** y decimos que corre en **tiempo polinomial**.

Los buenos algoritmos están en P. Aunque otros corren en **tiempo exponencial** que en la práctica significa que para algunas entradas el número de pasos puede ser muy grande porque dependen de una función exponencial.

Pueden llegar a ser intratables. En esos casos, suele ser mejor un algoritmo no determinista o probabilístico que sea polinomial.

Un **algoritmo probabilístico** es un algoritmo que basa su resultado en la toma de algunas decisiones. Así, se elige un elemento de un conjunto al principio del algoritmo o bien en cada iteración.

En promedio, obtienen una buena solución al problema planteado. Pero con la misma entrada se pueden obtener distintas soluciones y, en algunos casos, soluciones erróneas. Se dividen en dos tipos, Montecarlo y Las Vegas.

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

1.1. Tipo Montecarlo. Corren[1] en tiempo polinómico y producen una respuesta que es correcta con probabilidad >1/2. Se dividen en 3 clases:

Inclinados a Falso, cuando la respuesta es correcta cuando sea False.
Inclinados a Cierto, cuando la respuesta es correcta cuando sea True.
No inclinados, cuando la respuestas False o True pueden no ser correctas.

Ejemplos, los test de primalidad de Solovay-Strassen, el Baillie-PSW, el Miller-Rabin y en grupos la variante rápida del algoritmo de Schreier-Sims.

Por ejemplo, el test de Solovay-Strassen está inclinado a Cierto, porque contesta True si el input n es primo. Pero si el input n es compuesto, contesta False con probabilidad al menos 1/2 y contesta True con probabilidad menor que 1/2. Así, la respuesta False es correcta y la True puede no serlo.

Un algoritmo Montecarlo puede mejorarse si se ejecuta k veces ya que entonces la respuesta es correcta con probabilidad al menos $1-(1-\frac{1}{2})^k = 1-2^{-k}$. En el test de Miller-Rabin, se incrementa hasta $1-4^{-k}$.

En resumen, un algoritmo Montecarlo puede dar la respuesta correcta o bien una respuesta errónea con probabilidad baja.

[1]Introducidos en 1947, por Nicholas Metropolis 1915-1999, físico griego américano que trabajó en Los Álamos laboratorio donde se fabricó la primera bomba atómica. Fue bautizado así por su clara analogía con los juegos de ruleta de los casinos, el más célebre de los cuales es el de Montecarlo, casino cuya construcción fue propuesta en 1856 por el príncipe Carlos III de Mónaco, siendo inaugurado en 1861.

1.2. Tipo Las Vegas. Son algoritmos probabilísticos que siempre producen una respuesta correcta pero su tiempo es aleatorio con una media acotada polinómicamente. Se programan de forma que si en un tiempo prudencial, no encuentran la respuesta correcta se para y se informa del fallo.

Un algoritmo Las Vegas[2] no juega con la corrección sino con los recursos de computación. Para ciertos inputs puede correr en tiempo exponencial y en estos casos se suele parar la computación e informar del fallo.

Las Vegas puede usarse cuando el número de respuestas sea pequeño y su verificación sea relativamente fácil aunque calcular una solución sea difícil. El test de Lucas-Lehmer usado en el certificado de Prat es Las Vegas.

Igual que con un algoritmo de Montecarlo, la probabilidad de encontrar una solución correcta aumenta con el tiempo empleado en obtenerla y el número de veces que se ejecuta (elecciones diferentes).

Su esquema de implementación se parece al de los algoritmos de Montecarlo, pero se diferencian de ellos en que incluye una condición de parada si no se ha encontrado la solución correcta en un tiempo prudencial.

[2]Introducidos en 1979 por László Babai (1950 Budapest), profesor de ciencias de la computación de la Universidad de Chicago.

Enrique R. Aznar
Dpto. de Álgebra

2. Un test de Lucas-Lehmer

Dado $n \in \mathbb{N}$, si existe un $a \in \mathbb{Z}$ tal que $(a, n) = 1$ y su orden multiplicativo módulo n es $n-1$. O sea, si $a^{n-1} \equiv 1 \pmod{n}$ y todas las anteriores potencias son distintas de 1, entonces todos los naturales anteriores a n son primos con el porque son potencias, módulo n, de a que lo es. Hemos razonado que

Teorema 1. *Si $\exists a \in \mathbb{Z}$ tal que $O(a) = n-1$. Entonces, n es primo.*

El recíproco también es cierto ya que si p es primo, el grupo multiplicativo, $U(\mathbb{Z}_p)$, tiene tamaño $p-1$ y por ser abeliano finito, sabemos que existe un elemento cuyo orden es el mínimo común múltiplo de todos los órdenes. O sea, existe $a \in \mathbb{Z}$ tal que su orden multiplicativo, $O(a) = r$, módulo p es divisible por cualquier otro orden. Por tanto, $b^r \equiv 1 \pmod{p}$ para todo b primo con p y la ecuación $x^r - 1 = 0$ tiene $p-1$ raíces en $\mathbb{Z}_p$ que por ser un cuerpo no puede tener más raíces que su grado. O sea, $p-1 = r$, existe un **elemento primitivo** que genera a todos y $U(\mathbb{Z}_p)$ es un grupo cíclico.

Si $\exists a \in \mathbb{Z}$ tal que $a^{n-1} \equiv 1 \pmod{n}$ entonces $(a, n) = 1$ (ya que este mcd divide a 1), si además $a^r \not\equiv 1 \pmod{n}$ para todo divisor r de $n-1$ entonces $O(a) = n-1$. Si $n-1 = \prod q_j^{\beta_j}$, q_j primos, basta con $a^{(n-1)/q_j} \not\equiv 1 \pmod{n}$. Así, si conocemos la factorización de $n-1$, hemos demostrado que

Teorema 2. *[Lucas-Lehmer] Si $\exists a \in \mathbb{Z}$ tal que $a^{n-1} \equiv 1 \pmod{n}$ y $a^{(n-1)/q} \not\equiv 1 \pmod{n}$ para todo q divisor primo de $n-1$. Entonces, n es primo.*

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

3. Un elemento primitivo un certificado de primalidad.

En $\mathbb{Z}_5$, un elemento primitivo es el 2 (o su clase módulo 5) ya que su orden multiplicativo, módulo 5, es $4 = 5-1$. En efecto, como $5-1 = 4 = 2^2$, por

$$2^2 = 4 \equiv -1,\ 2^4 \equiv (-1)^2 = 1 \pmod 5$$

Lucas-Lehmer, basta con la primera potencia para certificar que 5 es primo.

Para los siguientes números de Fermat, $F_n = 2^{2^n}+1$, se tiene

$$2^{2^n} \equiv -1 \pmod{F_n} \Rightarrow 2^{2^{n+1}} \equiv 1 \pmod{F_n} \Rightarrow O(2) = 2^{n+1}$$

como $2^{n+1} < 2^{2^n} = F_n - 1$, salvo para $n = 1$ ($n = 2^{2^1}+1 = 5$ estudiado antes). Por tanto, 2 no puede ser un elemento primitivo para $F_n = 2^{2^n}+1$, con $n > 1$.

Como $2^2 = 4 \equiv 1 \pmod 3 \Rightarrow 2^{2^n} \equiv 1 \pmod 3$, el símbolo de Jacobi

$$\left(\frac{3}{F_n}\right) = \left(\frac{F_n}{3}\right) = \left(\frac{2}{3}\right) = -1$$

Cuando p es primo y $(a,p) = 1$, $a^{(p-1)/2} \equiv \left(\frac{a}{p}\right) \pmod p$. Luego, si F_n es primo $3^{(F_n-1)/2} \equiv -1 \pmod{F_n}$. Recíprocamente, si $3^{(F_n-1)/2} \equiv -1 \pmod n$, 3 es un elemento primitivo módulo y F_n es primo. Así, para $k > 1$[3]

Teorema 3. *[Pépin]* $n = 2^{2^k}+1$ *es primo si y sólo si* $3^{(n-1)/2} \equiv -1 \pmod n$.

[3] Jean Fraçois Théophile Pépin, en 1877, lo demostró para $a = 5$ en vez de $a = 3$.

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

Para el siguiente número de Fermat, $n = 17$, $n-1 = 16 = 2^4$, el 3 es primitivo

$$3^{16/2} = 3^8 \equiv 16 \equiv -1 \pmod{17}$$

Por Pépin o Lucas-Lehmer, esta congruencia certifica que 17 es primo.

Para el siguiente número de Fermat, $n = 257$, $n-1 = 256 = 2^8$, también

$$3^{256/2} = 3^{128} \equiv 256 \equiv -1 \pmod{257}$$

Por Pépin o Lucas-Lehmer, esta congruencia certifica que 257 es primo.

Para el siguiente número de Fermat, $n = 65537$, $n-1 = 65536 = 2^{16}$,

$$3^{65536/2} = 3^{32768} \equiv 65536 \equiv -1 \pmod{65537}$$

Por Pépin, esta congruencia certifica que 65537 es primo.

Para el siguiente número de Fermat, $F_5 = 2^{2^5} + 1 = 2^{32} + 1 = 4294967297$, $(F_5 - 1)/2 = 2147483648$,

$$3^{2147483648} = 10324303 \not\equiv -1 \pmod{4294967297}$$

Por Pépin, esta congruencia certifica que 4294967297 es compuesto.

Como la exponenciación rápida es muy eficiente, por el mismo método, se ha demostrado que todos los F_n para $n \in \{5, \ldots, 32\}$ son compuestos[4]. Todavía, es una cuestión abierta si F_0, F_1, F_2, F_3, F_4 son los únicos primos de Fermat.

[4] Otra cuestión, es encontrar sus factores primos. Todavía no se conocen todos.

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

4. Pseudoprimos de Fermat. Números de Carmichael.

Dado un entero positivo n, se elige un entero positivo pequeño a (una **base** para n), se calcula la potencia $a^{n-1} \mod n$, si no sale 1, n es compuesto[5]. Pero puede dar 1, siendo n compuesto. En ese caso, decimos que n es un **pseudoprimo** (de Fermat), respecto de la base a, **psp(a)**.

Para una base a fija, hay muchos menos psp(a) que números primos que pasan el test. Cuando un n pasa el test de Fermat, decimos que es un **Fermat posible primo**. Pero hay compuestos que son psp(a) para todo a primo con el. Lo que indica que este test puede fallar en detectar números primos.

Definición 1. ***Número de Carmichael*** *es un n compuesto tal que $a^{n-1} \equiv 1 \mod n$ para todo a con $mcd(a,n) = 1$.*

Ejemplo 1. *Con el algoritmo de la exponenciación rápida es fácil calcular potencias modulares. Así para $n = 27213609$ y para la base $a = 2$ se tiene*

$$2^{n-1} \mod n = 4$$

Como es distinto de 1, muestra que $n = 27213609$ es un número compuesto.

En cambio, para $n = 27213647$, se tiene que $a^{n-1} \mod n = 1$ para las bases $a = 2,3,5,7,11$. O sea, $n = 27213609$ es posible primo para esas 5 bases. Sin embargo, no podemos descartar que sea compuesto (por ej., Carmichael).

[5] Por el teorema pequeño de Fermat (TPF), si n es primo debería dar 1.

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

5. Pseudoprimos de Euler. Test de Solovay-Strassen

Si $n = 2m+1$ es un primo impar, el TPF (teorema pequeño de Fermat) dice que $a^{2m} = a^{n-1} \equiv 1 \mod n$. En este caso, $a^m = a^{(n-1)/2} \mod n$ es una raíz cuadrada de 1 y por tanto debe ser ± 1. Si no lo es n es compuesto[6].

En caso de dar 1 o -1, se calcula el símbolo de Jacobi $(\frac{a}{n})$. Si no coincide con el valor de la potencia anterior, n es compuesto[7]. En caso contrario, n es **Euler posible primo**. Los números compuestos que pasan este test, se llaman **pseudoprimos de Euler** respecto de la base a, **epsp(a)**.

Ejemplo 2. *Para el Fermat posible primo visto antes,* $n = 27213647$, *como* $(n^2-1)/2 = 370291291520304$ *es par, se tiene que*

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

O sea, $n = 27213647$ *es un Euler probable primo para* $a = 2$.

Como $(-1)^{(3-1)(n-1)/4} = (-1)^{13606823} = -1$ *se tiene que*

$$\left(\frac{3}{n}\right) = -\left(\frac{n}{3}\right) = -\left(\frac{2}{3}\right) = 1 = 3^{(n-1)/2} \mod n$$

También, $n = 27213647$ *es un Euler probable primo para* $a = 3$.

---

[6] Porque cuando n es primo, $\mathbb{Z}_n$ es cuerpo y $\mathbb{Z}_n[x]$ es un DE.

[7] Por el criterio de Euler: Si n es primo, $(\frac{a}{n}) \equiv a^{(n-1)/2} \mod n$.

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

6. Pseudoprimos fuertes. Test de Miller-Rabin

En el caso, $a^m = a^{(n-1)/2} \equiv 1 \mod n$ y $m = 2k$ fuera par, podemos aplicar el mismo razonamiento a la potencia $a^k = a^{(n-1)/4} \mod n$ que debe ser ± 1. Si no lo es, n es compuesto. Si sale 1 y k es de nuevo par, podemos repetir el argumento hasta llegar a un exponente impar.

Si $n-1 = 2^r m$ con m impar y a es una base pequeña, llamamos a la sucesión

$$a^m, a^{2m}, , a^{4m}, \ldots, a^{2^r m} = a^{n-1}$$

una **a-sucesión** para n. Si esta sucesión no acaba en 1 o bien hay un 1 precedido por un entero que no es ± 1, n es compuesto.

En caso contrario, n se comporta como primo. Decimos que n ha pasado el **test de pseudoprimos fuerte** para la base a. A un número n compuesto que pasa el test anterior lo llamamos **pseudoprimo fuerte** respecto de la base a.

A una base a para la que un n es pseudoprimo fuerte (pasa el test) la lamamos un **mentiroso fuerte** para el número compuesto n.

El número de mentirosos fuertes para un n compuesto no puede ser mayor que $(n-1)/4$ y es mucho menor en muchos casos.

Por tanto, la probabilidad de que un n compuesto pase el test sPs para m bases elegidas al azar es menor que 4^{-m}. Esta versión probabilística del test sPs se llama **test de Miller-Rabin**.

Enrique R. Aznar
Dpto. de Álgebra

El número de términos de cualquier a-sucesión es uno mas el número de ceros a la derecha en la expresión de n-1 en base dos. Para su cálculo se calcula la potencia modular con exponente impar (el más pequeño de la sucesión) y luego se calculan sus cuadrados modulares sucesivos.

Ejemplo 3. *Para* $n = 27213609$ *y cualquier base, la a-sucesión tiene 4 términos ya que* $n-1 = 27213608$, $(n-1)/2 = 13606804$, $(n-1)/4 = 6803402$ *y* $(n-1)/8 = 3401701$ *ya es impar. Para* $a = 2$, *esas potencias modulares son*

$$2^{3401701} \equiv 14728406, 2^{6803402} \equiv 6831766, 2^{13606804} \equiv 27213607, 2^{27213608} \equiv 4$$

Y como no acaba en 1 (acaba en 4), demuestra que 27213609 *es compuesto.*

Ejemplo 4. *Para el probable primo,* $n = 27213647$ *y cualquier base, la a-sucesión tiene sólo dos términos ya que* $(n-1)/2 = 13606823$ *es impar.*

Para $a = 2$, *la 2-sucesión es*

$$2^{13606823} \equiv 1, \quad 2^{27213646} \mod n \equiv 1$$

O sea, $n = 27213647$ *es un Miller-Rabin posible primo para* $a = 2$.

También, la 3-sucesión sale

$$3^{13606823} \equiv 1, \quad 3^{27213646} \mod n \equiv 1$$

Y $n = 27213647$ *tes un Miller-Rabin posible primo para* $a = 3$.

Estos cálculos no demuestran que $n = 27213647$ *sea primo, pero si nos dan una probabilidad muy alta de que lo sea (al menos* $1 - 4^{-2} = 0.9375$*).*

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

7. CERTIFICADOS

Un certificado de primalidad es una prueba formal simple de que un número natural es primo. Permite comprobarlo rápidamente sin tener que correr ningún otro test de primalidad costoso y probabilístico. Prueba simple significa que es polinómica en el número de cifras del número inicial n.

La existencia para todo n, de un certificado de primalidad conduce directamente a que el problema de comprobar la primalidad (complemento de la factorización entera) pertenece a la clase NP (nondeterministic polynomial time), la clase de problemas cuya solución se comprueba en tiempo polinomial con un algoritmo probabilístico.

Como el problema complemantario, la factorización entera, es claramente NP, decimos que comprobar la primalidad es co-NP. Por tanto, comprobar la primalidad está en la intersección NP con co-NP. Recientemente, se ha comprobado que en realidad este problema está en P[8].

Un certificado para el problema complementario es la exhibición de cualquier divisor de n. Algunos test probabilísticos de primalidad conocidos como el de Baillie-PSW, el de Fermat, y el de Miller-Rabin también producen certificados de composición de un número n, en caso de serlo, pero no producen ningún certificado de primalidad cuando es primo.

[8] Agrawal, Kayal y Saxena en agosto de 2002, publicaron un algoritmo determinista de clase P para la determinación de la primalidad de un número

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

8. El certificado de Pratt

El certificado de Pratt[9] es un certificado de primalidad basado en el lema de Lucas-Lehmer (un recíproco del teorema pequeño de Fermat). Antes del trabajo de Pratt, se consideraba un método heurístico que funcionaba en la mayor parte de los casos (Knuth 1969). Pratt en 1975, mostró que se podía usar como un procedimiento no determinista simplemente aplicándolo recursivamente a los factores de n-1. Fue el primero en demostrar que el árbol resultante del método recursivo implica que la factorización en primos pertenece a la clase de complejidad NP.

Para generar un certificado de Pratt, suponemos que n es un entero positivo y $\{p_i\}$ el conjunto de los factores primos de n-1. Llamamos **testigo de primalidad** a un entero x tal que $x^{n-1} \equiv 1 \mod n$ pero que $x^{(n-1)/p_i} \not\equiv 1 \mod n$.

El orden multiplicativo $\mod n$ de un testigo b divide a $n-1$ y necesariamente coincide con el. Por tanto, las primeras $n-1$ potencias de b módulo n dan todas las clases módulo n que son todos los números menores que n. Estos son primos relativos con n y este es primo (lema de Lucas-Lehmer).

Se puede mejorar algo la definición de **testigo**:

[9] Vaughan Pratt (nacido el 12 de abril de 1944) es un profesor emérito de la universidad de Stanford (USA), que fue pionero en ciencias de la computación. Ha hecho contribuciones a áreas fundamentales como algortimos de búsqueda, de ordenación, y test de primalidad.

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

Theorem 4. *Si existe un entero x tal que* $x^{(n-1)/2} \equiv -1 \mod n$ *pero que* $x^{(n-1)/2p_i} \not\equiv -1 \mod n$ *para todo* p_i *divisor primo impar de* $n-1$. *Entonces,* $ord_n(b) = n-1$ *y n es primo.*

Aplicando estos recíprocos del teorema pequeño de Fermat a n y a cada presunto factor primo de n-1, se puede generar un certificado de cada uno de estos primos. El certificado de Pratt proporciona un prueba de que un número es una raíz primitiva del grupo multiplicativo (mod n), junto con el hecho de que ese grupo tiene orden n-1, prueba que n is primo.

Ejemplo 5. $n = 7919$, *factorizamos* $n-1 = 7818 = 2*37*107$ *y probamos*

$$7^{7818} \equiv 1, 7^{7818/2} \not\equiv 1, 7^{7818/37} \not\equiv 1, 7^{7818/107} \not\equiv 1 \mod n$$

que 7 es un testigo de la primalidad de n=7919.

Además, recursivamente se comprueba que 2 es un testigo para 37 porque, $36 = 2^2 * 3^2$ *y módulo 37 se tiene*

$$2^{36} \equiv 1, 2^{36/2} \not\equiv 1, 2^{36/3} \not\equiv 1$$

También, 2 es un testigo para 107 porque, $106 = 2*53$ *y módulo 107 se tiene*

$$2^{106} \equiv 1, 2^{106/2} \not\equiv 1, 2^{106/53} \not\equiv 1$$

También, 2 es un testigo para 53 porque, $52 = 2^2 * 13$ *y módulo 53 se tiene*

$$2^{52} \equiv 1, 2^{52/2} \not\equiv 1, 2^{52/13} \not\equiv 1$$

También, 2 es un testigo para 13 porque, $12 = 2^2 * 3$ *y módulo 13 se tiene*

$$2^{12} \equiv 1, 2^{12/2} \not\equiv 1, 2^{12/3} \not\equiv 1$$

También, 2 es un testigo para 3 porque, $3 - 1 = 2$ *y módulo 3 se tiene*

$$2^2 \equiv 1, , 2^{2/2} = 2 \not\equiv 1$$

Finalmente, admitimos que 2 es primo sin demostración.

El proceso anterior es un árbol que se puede codificar linealmente. Así

$$\{7919, 7, \{2, \{37, 2, \{2, \{3, 2, \{2\}\}\}\}\}, \{107, 2, \{2, \{53, 2, \{2, \{13, 2, \{2, \{3, 2, \{2\}\}\}\}\}\}\}\}\}$$

La forma de hacerlo recursivamente es la siguiente. Comenzamos

$$\{7919, 7, \{2, 37, 107\}\}$$

Ahora, hacemos lo mismo para 2, 37 y 107 que son los presuntos factores primos de $7919 - 1 = 7918$*:*

$$\{2\}, \quad \{37, 2, \{2, 3\}\}, \quad \{107, 2, \{53, 2\}\}$$

Como 2 admitimos que es primo, quedan los presuntos factores 3 y 53

$$\{3, 2, \{2\}\}, \quad \{53, 2, \{13, 2\}\}$$

queda sólo certificar que 13 es primo

$$\{13, 2, \{3, 2\}\}$$

Finalmente, sustituimos sucesivamente desde el principio cada uno de los presuntos factores primos por su certificado.

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

9. El certificado de AKS

En agosto del año 2002, Agrawal, Kayal y Saxena de la Universidad de Kanpur presentaron un algoritmo determinista de clase P para la determinación de la primalidad de un número. La clave del algoritmo es una versión simplificada del TPF (teorema pequeño de Fermat) para polinomios modulares:

Teorema 5. . *Si* $2 < n, r \in \mathbb{Z}$ *con el orden de* n *en* $U(\mathbb{Z}_r)$ *mayor que* $\log_2^2 n$ *y*

$$(x-a)^p \equiv (x^p - a) \mod x^r - 1, p$$

para todo $0 \le a \le \sqrt{\varphi(r)}\log_2 n$. *Si* n *tiene un factor primo* p, *con* $\sqrt{\varphi(r)}\log_2 n < p$. *Entonces,* $n = p^m$. *En particular, si* n *no es una potencia de primo y no tiene factores primos en* $[1, \sqrt{\varphi(r)}\log_2 n]$, *entonces* n *es primo.*

La demostración es larga y se puede ver en [3], pag. 202.

Un algoritmo AKS, basado en el teorema, puede ejecutarse en un tiempo de

$$O((\log n)^{\frac{21}{2}} f(\log\log n))$$

El algoritmo tiene de entrada un n>1 y de salida COMPUESTO o PRIMO.

1) Si existen $a, b \in \mathbb{N}$ con $1 < b$, $n = a^b$, n es compuesto.

2) Encuentre el más pequeño valor de r tal que $O_r(n) > \log_2^2(n)$.

3) Si $\text{mcd}(a, n) \neq 1$, para algún natural $a \le r$ entonces n es compuesto.

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

4) Si $\sqrt{n} \leq r$ entonces n es primo.

5) Para $a = 1$ hasta $\sqrt{\varphi(r)} \log_2 n$: Si

$$(x-a)^n \not\equiv (x^n - a) \mod x^r - 1, n$$

entonces n es compuesto. Si se termina el ciclo sin salir antes, n es primo.

Esta versión del algoritmo (con un error en el paso 4) es la que aparece en https://en.wikipedia.org/wiki/AKS_primality_test

Allí, aparece como ejemplo la demostración de que 31 es primo, aunque sobra el paso 5) porque la constante $r = 29$ es mayor que $\sqrt{31} \simeq 5.56$[10].

El algoritmo AKS requiere un tiempo $O\left(\log_2^{21/2} n\right)$. Es decir, AKS es determinista y corre en un tiempo polinómico. En la práctica, el algoritmo parece correr en un tiempo $O\left(\log^6 n\right)$. Este tiempo se podría demostrar siempre y cuando se demuestre la conjetura de los números primos gemelos.

Si embargo, AKS no sirve para números relativamente pequeños. Para $n = 16493$, la constante r debe ser mayor que $\lfloor \log_2^2(n) \rfloor = 196$ que es mayor que la raíz cuadrada $\lfloor \sqrt{n} \rfloor = 128$ y de nuevo sobra el paso 5).

Comprobar si n es una potencia, paso 1), es fácil de hacer. En Mathematica:

m = Floor[Log[2, n]]; For[i = 2, i <= m, i++, b = Floor[$n^{1/i}$];

[10] El mismo error aparece en la versión en español de wikipedia, en febrero de 2018.

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

If[b^i == n, Print["n es una potencia"];Return[i]]];
Print["n no es una potencia"]; Return[0]

Sin embargo, en el mismo ciclo For, es más fácil hacer el mcd para comprobar que n por debajo de $m = \lfloor \log_2(n) \rfloor$ no tiene factores. Por ej., para $n = 16493$, $m = 14$. Para $n = 27213647$, $m = 24$. Ese ciclo es rapidísimo.

El cálculo de la constante r es fácil de programar. En Mathematica:

Module[m, r, s, i = 1, m = (Log[2., n])^2; r = NextPrime[m];
s = MultiplicativeOrder[n, r];
While[s <= m, If[Mod[n, r] == 0, Return["n compuesto"]]; i++;
r = NextPrime[m, i]; s = MultiplicativeOrder[n, r];];
Return[r]]

Finalmente, los pasos 3) y 4), se pueden programar juntos con un ciclo en el paso 5). Lo más costoso[11], es el cálculo de $(x-a)^n \not\equiv (x^n - a) \mod x^r - 1, n$.

Por ejemplo, para $n = 27213647$, la constante sale, $r = 643$, un orden de magnitud menor que la raíz cuadrada $\lfloor \sqrt{n} \rfloor = 5216$. Sin embargo, como el paso 5), trata con polinomios de grado 643, cada iteración tarda lo suficiente para cansar al que ejecuta y es sólo un paso de las 625 congruencias modulares necesarias para demostrar que 27213647 es primo con AKS.

[11] Aún con exponenciación rápida.

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

10. Algunas Implementaciones.

Mathematica tiene en su núcleo una función llamada **PrimeQ** que primero prueba divisores primos pequeños para n. Si no encuentra, usa el test de Miller-Rabin (de pseudoprimos fuertes) con base 2 y base 3. Si n pasa, usa un test de Lucas. Si pasa este último, devuelve True (que n es primo).

Con los conocimientos de 1997, PrimeQ es correcto para $n < 10^{16}$. O sea, es determinista en ese intervalo. Para n mayores la función es probabilítica.

FactorInteger de Mathematica prueba divisores pequeños. Después, alterna los métodos de Pollard, Pollard rho, curvas elípticas y criba cuadrática.

Maple tiene en su núcleo una función llamada **isprime** que primero usa el test de Miller-Rabin con 25 base elegidas aleatoriamente. Si n pasa estos tests, la probabilidad de que no sea primo es $4^{-25} = 8.88178 * 10^{-16} < 10^{-15}$. Finalmente, usa un test de Lucas. Si pasa este último, devuelve que n es primo. Así, la probabilidad de error de isprime es mucho menor de 10^{-15}.

ifactor de Maple usa por defecto el método de criba cuadrática múltiple polinomial. Pero se puede especificar en un segundo parámetro el método de factorización a usar[12].

[12] Fracciones continuas de Morrison and Brillhart, Curvas elípticas de Lenstra, Pollard rho o Factorización libre de cuadrados de Shanks.

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

11. Referencias.

[1] David Bressoud, Stan Wagon: *A Course in Computational Number Theory*, John Wiley & Sons, Hoboken, NJ, USA, 2000.
[2] H. Cohen: *A Course in Computational Algebraic Number Theory*, vol 138 Graduate Texts in Mathematics. Springer-Verlag, 2000.
[3] Richard Crandall, Carl B. Pomerance: *Prime numbers. A computational perspective*, Springer Science+Business Media, Inc., USA, 2005.
[4] Hans Riesel: *Prime Numbers and Computer Methods for Factorization*, Springer Science+Business Media, LLC 2012, (first edition Birkhäuser, 1994).
[5] Ian Stewart, David Tall: *Algebraic Number Theory and Fermats Last Theorem*, AK Peters, Ltd, USA, 2002.
[6] Samuel S. Wagstaff, Jr: *The joy of factoring*, AMS, Providence, Rhode island, 2013.

12. Test de repaso.

Para comenzar el cuestionario pulsa el botón de inicio.
Cuando termines pulsa el botón de finalizar.
Para marcar una respuesta coloca el ratón en la letra correspondiente y pulsa el botón de la izquierda (del ratón).

Inicio del Test

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

1. ¿Cuál de las siguientes afirmaciones es la correcta?.

(a) Un test probabilístico tipo Montecarlo corre en tiempo exponencial pero su respuesta es correcta con probabilidad >1/2.

(b) Un test probabilístico tipo Montecarlo corre en tiempo polinomial pero la respuestas siempre es correcta cuando es True.

(c) Un test probabilístico tipo Montecarlo corre en tiempo polinomial pero la respuestas siempre es correcta cuando es False.

(d) Un test probabilístico tipo Montecarlo corre en tiempo polinomial y puede ser inclinado a True, inclinado a False o no inclinado.

2. ¿Cuál de las siguientes afirmaciones es verdadera?.

(a) Un test probabilístico tipo Las Vegas juega con la corrección de las respuestas no con los recursos de computación.

(b) Un test probabilístico tipo Las Vegas puede producir una respuesta errónea.

(c) Un test probabilístico tipo Las Vegas siempre produce una respuesta correcta en tiempo polinomial.

(d) Un test probabilístico tipo Las Vegas produce una respuesta correcta en tiempo aleatorio cuya media está acotada polinomialmente.

3. ¿Cuál de las siguientes afirmaciones es verdadera?.

(a) Un teorema de Lucas-Lehmer dice que en $\mathbb{Z}_n$ siempre existe un elemento primitivo.
(b) Un número entero n es primo cuando el grupo aditivo $\mathbb{Z}_n$ sea un grupo cíclico.
(c) Un teorema de Lucas-Lehmer dice que n es primo si en $\mathbb{Z}_n$ existe una clase de orden multiplicativo máximo.
(d) Existe un número entero n no primo con el grupo multiplicativo de las unidades módulo n cíclico.

4. ¿Cuál de las siguientes afirmaciones es verdadera?.
(a) El test de Pépin para números de Fermat puede correr en tiempo exponencial.
(b) El test de Pépin es muy eficiente para hallar factores de un número de Fermat.
(c) No existe ningún test eficiente para comprobar la primalidad de un número de Fermat.
(d) Aunque sea fácil comprobar la primalidad de un número de Fermat puede ser difícil demostrar la primalidad de alguno de sus factores.

5. ¿Cuál de las siguientes afirmaciones es verdadera?.
(a) Un pseudoprimo de Fermat, respecto de la base a, n = psp(a), es siempre un número de Carmichel.

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

(b) Un número de Carmichel puede no ser psp(a) para alguna base a.
(c) Un pseudoprimo de Fermat, n = psp(a), respecto de la base a, es simplemente un entero primo con a.
(d) Un pseudoprimo de Fermat, n = psp(a), satisface la congruencia $a^{n-1} \equiv 1 \pmod{n}$ y es compuesto.

6. ¿Cuál de las siguientes afirmaciones es verdadera?.
(a) Existen infinitos números de Carmichel porque existen infinitos primos.
(b) Existen infinitos pseudoprimos de Fermat porque existen infinitos números de Carmichel.
(c) Existen sólo un número finito de pseudoprimos de Fermat y de números de Carmichel.
(d) Sólo se conocen un número finito de números de Carmichel.

7. ¿Cuál de las siguientes afirmaciones es verdadera?.
(a) Un pseudoprimo de Euler respecto de la base a, n = epsp(a), es un número primo que parece compuesto.
(b) Un pseudoprimo de Fermat nunca es de Euler.
(c) Un pseudoprimo de Euler es un número compuesto que satisface la congruencia $a^{(n-1)/2} \equiv 1 \pmod{n}$.

(d) Un pseudoprimo de Euler respecto de la base a, n = epsp(a), es siempre un pseudoprimo de Fermat, respecto de la misma base.

8. ¿Cuál de las siguientes afirmaciones es verdadera?.

(a) Un pseudoprimo fuerte n para la base a, satisface que $a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \pmod{n}$.

(b) Un pseudoprimo fuerte n para la base a, nunca es un pseudoprimo de Euler para la misma base.

(c) Un pseudoprimo fuerte n para la base a, nunca es un pseudoprimo de Fermat para la misma base.

(d) Los pseudoprimos fuertes pueden certificar que un número es compuesto pero no que es primo.

9. ¿Cuál de las siguientes afirmaciones es verdadera?.

(a) El test de Solovay-Strassen certifica la primalidad pero el de Miller-Rabin sólo la composición.

(b) El test de Solovay-Strassen certifica que un número es compuesto pero el de Miller-Rabin sólo la primalidad.

(c) Los tests de Solovay-Strassen y el de Miller-Rabin pueden certificar la primalidad de un número.

(d) Los tests de Solovay-Strassen y el de Miller-Rabin pueden certificar que un número es compuesto.

ugr Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

Universidad de Granada
Enrique R. Aznar
Dpto. de Álgebra

10. ¿Cuál de las siguientes afirmaciones es verdadera?.

(a) El certificado de Pratt se aplica cuando sabemos que un número es compuesto.

(b) El certificado de Pratt es recursivo y usa sucesiones de Lucas.

(c) El certificado de Pratt es recursivo y sólo usa el teorema pequeño de Fermat.

(d) El certificado de Pratt es recursivo y usa el concepto de orden multiplicativo módulo n.

Final del Test | Puntos: