Def: n impar, $n-1 = 2^k q$, q impar. $a \in \mathbb{Z}$ t.q
$\text{mcd}(a,n) = 1$ es un testigo Miller Rabin si

1) $a^q \equiv 1 \pmod n$
2) $a^{2^i q} \not\equiv -1 \pmod n$ $\forall\ i \in \{0, 1, 2, \dots, k-1\}$

Obs: si existe tal $a \Rightarrow n$ es compuesto

¿Porque Miller-Rabin > Fermat?
No hay números Carmichael para esta prueba.

Prop: n compuesto e impar. Entonces > 75% de los numeros a entre 1 y $n-1$ son testigos Miller Rabin.

Rabin Miller algoritmo:

1) elegir $a_1, a_2, \dots, a_k$ ~~independientes~~ distintos $1 < a_i < n$
2) ver si cada uno es un testigo uno
3) si alguno es un testigo $\Rightarrow$ n compuesto
4) si ninguno es compuesto $\Rightarrow$ n probablemente primo

Si $k = 100$, la probabilidad que lo ~~sea~~ adivinamos mal es

$$(0.25)^{100} < 10^{-60}$$

¿Qué probable es que un número sea primo? (16)

Def: $\pi(x) = \#\{p \le x : p \text{ primo}\}$

Teo: $\lim_{x\to\infty} \frac{\pi(x)}{x/\log x} = 1$

En general usamos primos de 1024 bits $\approx 10^{300}$ b-B.

¿Cuántos primos hay de ese tamaño?

$$\pi(2^{1024}) - \pi(2^{1023}) \approx \frac{2^{1024}}{\ln 2^{1024}} - \frac{2^{1023}}{\ln 2^{1023}}$$

$$\approx 2^{1013.5}$$

La probabilidad que N es primo es $\boxed{\frac{1}{\ln N}}$

(porque $\frac{\text{resultados favorables}}{\text{resultados}} = \frac{x/\log x}{x} = \frac{1}{\log x}$.

Ej: Un número de 1024 bits tiene $\frac{1}{\log 2^{1024}} \approx 0.14\%$ probabilidad de ser primo. Entonces

tendríamos que elegir ≈ 700 números al azar para encontrar un primo.

Se puede mejorar. Consideramos el primorial

$p\#$ = producto de los primos hasta p.

~~Lors~~ y $11\# = 2310$ en particular. Consideramos para varios K

$K\,11\# + \underbrace{1139}_{\text{es primo con } 11\#}$.

La probabilidad que $N = 2310K + 1139$ es primo es:

$$\frac{1}{\frac{1}{2}\frac{2}{3}\frac{4}{5}\frac{6}{7}\frac{10}{11}} \cdot \frac{1}{\ln N} \approx \frac{4.8}{\ln N}$$ y para

$N \approx 2^{1024}$ es 0.67%.

probabilidad que → impar es primo $\frac{2}{\log N}$, $\frac{\frac{3}{2}}{\log N}$ Z/(3) primo

Entonces eligiendo 150 números al azar debería resultar en un primo.

Usamos MR con 100 posibles testigos para verificar si es primo.

<u>Demostración vs prueba</u>:

MR encuentra primos probables; no encuentra números garantizados ser primos.

<u>Trial division</u>: dividir por todo ~~enter~~ primo hasta $\sqrt{n}$ ~~prueba~~ demuestra ~~si~~ definitivamente si n es primo o no.

→ <u>muy</u> lento, algoritmo exponencial.

La Hipótesis de Riemann:

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} \quad s \in \mathbb{C} \; : \quad \text{Converge si } Re(s) > 1.$$

$$\sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_{p \text{ primo}} \frac{1}{1-p^{-s}}$$ ~~$\Rightarrow \zeta(z) = \# \{$~~

Teo: $\zeta(s)$ tiene una continuación analítica a todo $\mathbb{C}$ salvo polos en $s = 0$ y $s = 1$. Tiene ceros triviales en $s = -2, -4, -6, \ldots$

La Hipótesis de Riemann: todo cero no trivial tiene $Re(s) = \frac{1}{2}$

$$\Updownarrow$$

$$\pi(x) = \int_2^x \frac{dt}{\ln t} + O(\sqrt{x} \ln x)$$

$$\pi(x) \sim \frac{x}{\log x} \iff \zeta(s) \text{ no tiene ceros en } Re(s) = 1.$$

La HR se puede generalizar a ~~otras~~ una clase de funciones llama de funciones L.

Prop: Si esta generalización de RH vale, cada compuesto n tiene un testigo Miller Rabin a tal que

$$a \leq 2 (\ln n)^2.$$

Cor: ~~Usando~~ Si GRH es verdad, se puede usar Miller-Rabin para probar que un número es primo (en tiempo polinomial)

¿Y si no queremos ~~usar~~ asumir una conjetura?

Thm: (AKS prueba de primalidad)

$\forall \varepsilon > 0$, $\exists$ algoritmo que definitivamente determina si n es primo en no más que $O((\ln n)^{6+\varepsilon})$ pasos.

Obs: mucho más lento que MR.

Pollard Método de $(p-1)$ de Pollard para factorizar:

Dado p es fácil determinar si p es primo. $\Rightarrow$ bueno para RSA

Pero RSA depende de la dificultad de factorizar.

El método $(p-1)$ de Pollard nos deja factorizar cierta clase de números rapidamente.

Algoritmo: Recibe N para factorizar.

1. $a \leftarrow 2$
2. for $j = 2, 3, 4, \ldots$, una cota
3. $a \leftarrow a^j \bmod N$
4. $d = \gcd(a-1, N)$
5. si $1 < d < N$ return d.
6. incrementar j.

¿Por qué funciona?

Supongamos que recibimos $N = pq$ y encontramos L t.q

$$p-1 \mid L \quad \text{y} \quad q-1 \nmid L.$$

O sea, $L = i(p-1)$ y para todo $L = i(q-1) + k$.

Como Si a es pequeño, p, q grande, ~~usando~~ calculamos ~~(usando~~ a^L usando Fermat: (es probable que $\text{mcd}(a, p, q) = 1$.

$$a^L = a^{i(p-1)} \equiv 1 \pmod{p}$$
$$\equiv a^k \pmod{q}.$$

Es improbable que $a^k \equiv 1 \pmod{q}$. Entonces para casi todo a

$$p \mid a^L - 1 \quad \text{y} \quad q \nmid a^L - 1.$$

Entonces $$p = \text{mcd}(a^L - 1, N).$$

¿Cómo se encuentra L?

Si $p-1$ es el producto de muchos primos pequeños, entonces ~~tal~~ $(p-1) \mid n!$ para un n no tan grande.

Entonces para cada $n = 2, 3, 4, \ldots$ calculemos $\text{mcd}(a^{n!} - 1, N)$ El $n!$ es el L.

Obs: 1) Como solamente tenemos que calcular $\text{mcd}(a^{n!}-1, N)$ reducimos $a^{n!}-1 \bmod N$ en cada paso

2) usamos $a^{(n+1)!} = (a^{n!})^{n+1} \pmod N$

3) $a^{n!} \pmod N$ se puede calcular en $2n \log_2 n$ pasos

$n! \approx \left(\frac{n}{e}\right)^n$ por Stirling

y exponenciación rápida.

4) es fácil de evitar tal p y q cuando uno genera p y q para RSA.

5) ¿Qué es la probabilidad de un entero $\approx n$ elegido al azar divide $B!$?

Tiene que ver con números B-lisos. Regresaremos a eso.

Factorizando usando diferencia de cuadrados.

$$X^2 - Y^2 = (X+Y)(X-Y).$$

Empezamos con N y sumamos b^2 para $b = 1, 2, \ldots$ hasta que $N + b^2$ es un cuadrado perfecto. Entonces

$$N + b^2 = a^2 \Rightarrow N = (a+b)(a-b).$$

¿Cómo se puede encontrar b más eficientemente?

Antes de empezar con N se puede empezar con cualquier ~~factor~~ múltiplo de N

~~$kN = (a+b$~~ $\quad kN + b^2 = a^2$

$\Rightarrow kN = (a+b)(a-b).$

Ahora $\text{mcd}(N, a+b)$ y $\text{mcd}(N, a-b)$ son factores de N

~~¿Cuántos k se puede elegir?~~

¿Qué pasa si lo pensamos módulo N?

$kN \equiv 0 \pmod N$ entonces buscamos soluciones $x^2 \equiv b^2 \pmod N$

Evitar de buscar a y b directamente, hacemos:

Procedimiento:

1. Crear relaciones:

encontrar muchos enteros $a_1, a_2, \dots, a_r$ t.q

cada $c_i \equiv a_i^2 \pmod N$

es el producto de primos pequeños

2. Eliminación

Tomar un producto $c_{i_1} c_{i_2} \cdots c_{i_s}$ de algunos de los $c_i = a_i^2$

$\Rightarrow$ tq cada primo que aparece en el producto tiene potencia par $\Rightarrow$ $c_{i_1} \cdots c_{i_s} = b^2$.

3. Calcular mcd: Sea $a := a_{i_1} a_{i_2} \cdots a_{i_s}$,

$d = \text{mcd}(N, a-b)$.

Como $a^2 = (a_{i_1} \cdots a_{i_s})^2 = c_{i_1} \cdots c_{i_s} = b^2$

es probable que d es un factor no trivial de N

Obs: ¿Cuántas soluciones $a^2 \equiv b^2 \pmod N$ tendremos que probar antes de factorizar N?

El caso más difícil es $N = pq$ para $p \neq q$.

Si encontramos a y b ~~aleatorios~~ al azar t.q $a^2 \equiv b^2 \pmod N$, ¿qué es la probabilidad que $\gcd(N, a-b)$ nos de un factor de N no trivial?

Sabemos que:

$$(a-b)(a+b) = a^2 - b^2 = kN = kpq$$

p tiene que dividir por lo menos uno de $a-b$ y $a+b$ y tiene la misma probabilidad para cada ~~uno~~ uno. Igual para q.

Ganamos si $a-b$ es divisible por exactamente uno de p o q y eso pasa ~50% del tiempo. Entonces si podemos generar situaciones a y b eficientemente, ~~entonces~~ se puede encontrar p o q.

Hay pasos de procedimiento:

1. Crear relaciones
2. Eliminación
3. mcd

Paso (3) es $\Theta(\ln N)$.

Hablamos de paso 4 de la próxima clase, pero ¿qué se puede decir (24) del paso 2?

En completar paso 1 cada $c_i = a_i^2 \pmod N$ es el producto de primos en el conjunto $\{p_1, p_2, \dots, p_t\}$ O sea, existe e_{ij} t.q.

$$c_1 = p_1^{e_{11}} p_2^{e_{12}} p_3^{e_{13}} \cdots p_t^{e_{1t}}$$
$$c_2 = p_1^{e_{21}} p_2^{e_{22}} p_3^{e_{23}} \cdots p_t^{e_{2t}}$$
$$\vdots$$
$$c_r = p_1^{e_{r1}} p_2^{e_{r2}} p_3^{e_{r3}} \cdots p_t^{e_{rt}}$$

Queremos encontrar $u_1, u_2, \dots, u_r \in \{0,1\}$ tq

$$c_1^{u_1} c_2^{u_2} \cdots c_r^{u_r} = \square$$

O sea

$$\prod_{i=1}^{r} c_i^{u_i} = \prod_{j=1}^{t} p_j^{\sum_{i=1}^{r} e_{ij} u_i} \quad \text{t.q} \quad \sum_{i=1}^{r} e_{ij} u_i \equiv 0 \pmod 2$$

O sea

$$\begin{pmatrix} e_{11} & e_{21} & e_{31} & \cdots & e_{r1} \\ e_{12} & e_{22} & e_{32} & \cdots & e_{r2} \\ \vdots & & & & \\ e_{1t} & e_{2t} & e_{3t} & \cdots & e_{rt} \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_r \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \pmod 2$$

(dado) (incógnitas)

$t \times r$ $\quad r \times 1$ $\quad t \times 1$

Paso (2) se reduce a álgebra lineal sobre $\mathbb{F}_2$ con matrices esparsas. y se puede hacer eficiente.