

Factorization Techniques from Fermat to Today

Universidad de Cuenca

Optativa 6 - Criptología

Dr. Diego Ponce

Capítulo 5

Freddy L. Abad L.

freddy.abadl@ucuenca.edu.ec

CAP 5

S.1 La entrada n del algoritmo de Fermat es importante porque siempre se calcula con n iguales a un primo, si colocamos un n for el algoritmo no realizará su propósito. En este caso no se considera al $\# 2$, que es for y además n se forma a partir de $x^2 - y^2 = n$

S.2

Aproximadamente $\sqrt{2}$

S.3

Si p & q son primos y $p < q$ no es necesario hacer la llamada a la función y_loop por lo tanto la familia que se utilizaría sería la función x_loop

$$\begin{aligned} r &= r + \mu \\ \mu &= \mu + 2 \end{aligned} \quad \left. \begin{array}{l} \text{se calcularía los factores} \\ \text{para la fórmula } n = pq. \end{array} \right.$$

S.4

Si $r \leq 0$ entonces

$$r = r + 4 \quad \mu = 2\sqrt{n} + 1$$

$$r + 2\sqrt{n} + 1 \geq 0$$

$$2\sqrt{n} \geq -r - 1$$

$$2\sqrt{n} \geq -(r+1)$$

$$\sqrt{n} \geq \frac{-(r+1)}{2}$$

↓

la raíz den n nunca puede ser < 0 por lo que al aumentar "r" $\sqrt{n} + 1$ se prueba que r no podrá ser negativo.

S.5

en el algoritmo S.1 ¿Porque no es necesario restar y volver a 0 cuando se sale de la función y_loop ?

- No es necesario realizar esto porque r al ser mayor a 0 siempre devuelven los valores actuales de r y y
 $r > 0$ siempre se cumple dentro de esta función.

S.6

$x^2 - n \rightarrow$ No es cuadrado perfecto en y_loop
 Entonces $r \neq 0$ cuando termine el bucle

$$n \neq y^2$$

$$n = r \rightarrow$$
 En el algoritmo

$$\begin{aligned} r > 0 &\rightarrow r - 1 \\ &\rightarrow \sqrt{r} \neq r \\ \therefore r &\neq 0 \end{aligned}$$

S.7

loop n

a = n

b = (C(n+1)/2)

while (b < a)

a = b

b = ((a*a + n)/(2*a));

System.out.println(a)

> 99?

n = 2

a = 2

b = 1

1 < 2 ✓

a = 1

b = 1,5

q = 1

$\sqrt{2} = 1,41 \vee$

i. a es el menor entero de \sqrt{n} LQD

ii. a puede ser mayor entero de \sqrt{n} dependiendo de la entrada n LQD

S.8

$\text{sqrt} \rightarrow \sqrt{n}$

r > 0

↓

r decrementa de
uno en uno

$\text{sqrt} \leftarrow \sqrt{n}$

r < 0

↑

r incrementa en

- 2 $\sqrt{n} + 1$

(1+2) < 2 \sqrt{n}

S.9

n = 19931831

factores = 3337, 6563

n = 342583509

factores = 13757, 28537

n = 2451839867

factores = 36947, 66361

n = 2786302931

factores = 37571, 74161

n = 13208340509

factores = 96493, 137713

antilogaritmo de 7 = 4

7 - 7 ← 0K7

otra

S.10

$n \rightarrow$ compuesto

" x " y " y " en teros randomicos satisface

$$x^2 \equiv y^2 \pmod{n}$$

$$n = x^2 - y^2$$

$$x^2 - y^2 \equiv 1 \pmod{n}$$

$$x^2 \equiv y^2 \pmod{n}$$

$$n = (x-y)(x+y)$$

$$1 = \frac{(x-y)}{n}$$

$$1 \neq \frac{(x+y)}{n}$$

... Puede tener 50% de probabilidad de que $x-y$ sea factor de n y 50% de que no.

S.11

Existen ≈ 100.000 numeros primos hasta el 2.000.000 y si los guardamos en memoria y realizamos la division trial., el algoritmo tardaria entre 1 - 10 segundos, en encontrar los factores de 1888129.

S.12

$$y_n = f(y_{n-1}) \pmod{d}$$

Hay numeros finitos de congruencia

para $x_i \equiv f(x_{i-1}) \pmod{n}$, y_i es congruente (\equiv)

con $f(y_{i-1}) \pmod{n}$

Por ejemplo:

$$\text{Si } y_0 = 2 \rightarrow n=1 \rightarrow f = x^2 + 1$$

$$y_0 = 2 \pmod{11} = 2$$

$$y_1 = 5 \pmod{11} = 5$$

$$y_2 = 10 \pmod{11} = 10$$

$$y_3 = 17 \pmod{11} = 5$$

$$y_4 = 26 \pmod{11} = 4$$

$$y_5 = 37 \pmod{11} = 4$$

$$y_6 = 49 \pmod{11} = 5$$

Se repiteadicamente
y de ahi tiene un
numero finito de
congruencias

5.13

Dado $x_i \equiv f(x_{i-1}) \pmod n$, y_i es congruente a $f(y_{i-1}) \pmod d$

↓

De esto se obtiene un finito de clases de congruencias entonces es por eso que

$$y_i = y_j$$

Entonces y_i se ve como un círculo con marcas y como $y_i = y_j$ entonces implica que

$$x_i \equiv x_j \pmod m.$$

5.14

$n \rightarrow$ compuesto

$$1 \leq r \leq n$$

$$g = \text{mcm}(r, n)$$

Si $g=1$ o $g=n$ se escoge otro random y se repite cuando otro no es trivial.

Difiere con el algoritmo 5.2 porque no termina cuando $g=n$, ademas queremos se da un máximo en un rango

1888129 es el menor numero primo dividiendo n

El # de ciclos es $\sqrt{1888129} = 1379$ ciclos

5.15

$y_0 = 2$	$502 = 5$	* Se realizaron 20 iteraciones de los 200 solicitadas
$y_1 = (2^2 + 1) \pmod{502}$	$503 = 26$	
$y_2 = (26^2 + 1) \pmod{503}$	$504 = 173$	
$y_3 = (173^2 + 1) \pmod{504}$	$505 = 135$	
$y_4 = (135^2 + 1) \pmod{505}$	$506 = 10$	
$y_5 = (10^2 + 1) \pmod{506}$	$507 = 101$	
$y_6 = (101^2 + 1) \pmod{507}$	$508 = 42$	
$y_7 = (42^2 + 1) \pmod{508}$	$509 = 238$	* El valor del índice del primer valor repetido es 18
$y_8 = (238^2 + 1) \pmod{509}$	$510 = 35$	
$y_9 = (35^2 + 1) \pmod{510}$	$511 = 204$	
$y_{10} = (204^2 + 1) \pmod{511}$	$512 = 145$	
$y_{11} = (145^2 + 1) \pmod{512}$	$513 = 506$	* El valor del índice final menos el inicial del repetido ($18 - 5 = 13$)
$y_{12} = (506^2 + 1) \pmod{513}$	$514 = 65$	
$y_{13} = (65^2 + 1) \pmod{514}$	$515 = 106$	
$y_{14} = (106^2 + 1) \pmod{515}$	$516 = 401$	
$y_{15} = (401^2 + 1) \pmod{516}$	$517 = 15$	
$y_{16} = (15^2 + 1) \pmod{517}$	$518 = 226$	
$y_{17} = (226^2 + 1) \pmod{518}$	$519 = 10$	
$y_{18} = (10^2 + 1) \pmod{519}$	$520 = 101$	
$y_{19} = (101^2 + 1) \pmod{520}$	$521 = 101$	
$y_{20} = (101^2 + 1) \pmod{521}$	$522 = 101$	

5.16 785994771147 → es primo

$$95016113332419 \rightarrow (882883)(1076203)$$

$$2506741191739 \rightarrow (910099)(275436)$$

$$227793195071137 \rightarrow (14134177) \times (16116481)$$

$$265870264098379 \rightarrow (12084661) \times (22000639)$$

5.18

$10.000! \approx 2720$ dígitos (aprox).

5.19 $10000/2 = 5000$

$$5000/2 = 2500$$

$$2500/2 = 1250$$

$$1250/2 = 625$$

$$625/2 = 312,5 \rightarrow$$
 Pongo la parte entera

$$312/2 = 156$$

$$156/2 = 78$$

$$78/2 = 39$$

$$39/2 = 19$$

$$19/2 = 9$$

$$9/2 = 4$$

$$4/2 = 2$$

$$2/2 = 1$$

5.20 $10000! = p-1 (q-1)(q-2)(q-3)\dots(1)$

$$(p-1) = (q-i) \rightarrow$$
 Se elimina el factor

$$p = (q-i+1) = 1000 + 1 - i = 10001 - i$$

$$\frac{2^{10000!} - 1}{p} = \frac{((2^i)^3)^{10000}}{10001 - i} - 1 \equiv 1 \pmod{p}$$

Como $p-1$ divide a $10000!$

$$m^{2^{10000!}} \pmod{n} \rightarrow m \equiv 1 \pmod{p}$$

S.21

El algoritmo S.2 necesita + ciclos para encontrar el factor primo ya que en el S.3 ya se conocen los factores primos y por ende ya no realiza una mayor cantidad que el S.2, el cual si busca otros factores a lo largo de su ejecución.

S.22

$m^{100000} \bmod n \rightarrow$ Es el factor mas largo

S.23

Usando el Alg. S.3 factorice los números del S.16

$$785994771137 \\ \hookrightarrow \text{Cs } \# \text{ primo}$$

$$265870264098379 \\ 12084661, 22000639$$

$$950161333249 \\ * 882803, 1076203$$

$$2506741191734 \\ \rightarrow \text{Cs } \# \text{ primo}$$

Los tiempos de ejecución no son muy variados ya que al venir en microsegundos (μs) no hay mucha diferencia significativa entre los tiempos de cada número.

$$227793195071137 \\ \rightarrow 14134177 * 16116481$$

S.24

INCOMPUTABLE