# Criptografia FIB

## 7. Primalitat

Anna Rio

Departament de Matemàtica Aplicada II • Universitat Politècnica de Catalunya

# Nombres primers

Un nombre enter $p \geq 2$ és primer si els seus únics divisors són $\pm 1, \pm p$

- Quants nombres primers hi ha?

  Infinits

  Prova d'Euclides: $p_1 p_2 \ldots p_n + 1$ té factors primers diferents de $p_1, \ldots, p_n$

$$
\begin{aligned}
2 + 1 &= 3 \\
2 \cdot 3 + 1 &= 7 \\
2 \cdot 3 \cdot 5 + 1 &= 31 \\
2 \cdot 3 \cdot 5 \cdot 7 + 1 &= 211 \\
2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 + 1 &= 2311 \\
2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 + 1 &= 59 \cdot 509 \\
2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 + 1 &= 19 \cdot 97 \cdot 277
\end{aligned}
$$

# Nombres primers

- Com estan distribuïts?
  Si $\pi(n)$ indica el nombre de primers menors o iguals que $n$, aleshores

$$\lim_{n\to\infty} \frac{\pi(n)}{n/\log n} = 1$$

| $n$ | $10^2$ | $10^3$ | $10^4$ | $10^6$ | $10^7$ |
|---|---|---|---|---|---|
| $n/\log n$ | 21.7 | 144.8 | 1085.7 | 72382.4 | 620421 |
| $\pi(n)$ | 25 | 168 | 1229 | 78498 | 664579 |

# Nombres primers: densitat

- En el conjunt dels enters positius menors o iguals que $n$, la proporció dels que són primers és el quocient $\dfrac{\pi(n)}{n}$
- Aquest quocient mesura la probabilitat que un nombre aleatori de l'interval d'enters $[1, n]$ sigui primer.
- També mesura la densitat dels primers en el conjunt dels enters positius.
- Si ens restringim al conjunt dels enters positius *senars*, la funció de densitat seria $\dfrac{\pi(n)}{n/2} = \dfrac{2\pi(n)}{n}$ i s'aproxima per $\dfrac{2}{\log n}$

| $n$ | $10^3$ | $10^{155}$ | $2^{1024}$ |
|---|---|---|---|
| $\dfrac{2}{\log n}$ | $\dfrac{1}{3.45}$ | $\dfrac{1}{178}$ | $\dfrac{1}{355}$ |

Hi ha bastants nombres primers

# CERCA DE NOMBRES PRIMERS

Si disposem d'un certificat o un test de primalitat, una manera raonable de fer cerca de nombres primers és

- $N$ aleatori (senar) (de longitud fixada)
- Passar el test a $N$, $N+2$, $N+4, \ldots$ fins obtenir resposta afirmativa

| | |
|---|---|
| N = Random [ Integer, $\{10^2, 10^3\}$ ] | 472 |
| N = N - 1 + N mod 2 | 471 |
| While[ !PrimeQ[N], N = N + 2] | 471 |
| | 473 |
| | 475 |
| | 477 |
| | 479 |

El cas pitjor faria 10 iteracions: 887 i 907 són primers consecutius.

# GARBELL D'ERATÒSTENES

## 230 BC

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |

**Eratosthenes of Cyrene**
Born 276 BC in (now) Libya
Died 194 BC in Alexandria

Eratosthenes develops Sieve to Find all Prime Numbers

# GARBELL D'ERATÒSTENES

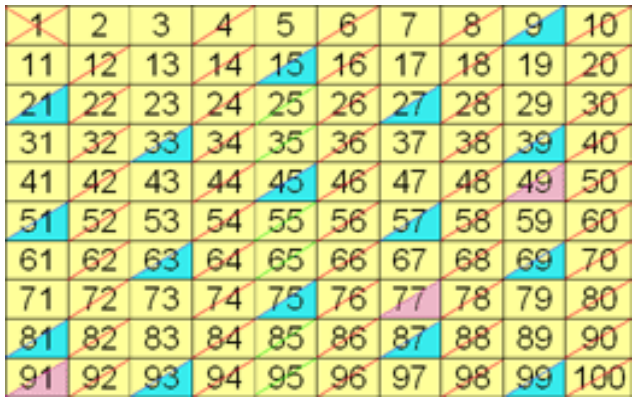
$N = ab \Rightarrow$ algun dels dos factors és $\leq \sqrt{N}$

## Taula de primers $\leq N$

- Inicialitzar la llista $\{2, 3, 4, 5, 6, 7, \ldots, N-1, N\}$
- El primer element de la llista és guarda com a primer, s'eliminen tots els seus múltiples.
- Això és repeteix fins que el primer element de la llista és $> \sqrt{N}$. Llavors tot el que resta són primers

| | | |
|---|---|---|
| 2 | $\{3, 5, 7, 9, 11, \ldots, 99\}$ | longitud 49 |
| 3 | $\{5, 7, 11, 13 \ldots, 97\}$ | longitud 32 |
| 5 | $\{7, 11, 13, 17, \ldots, 97\}$ | longitud 25 |
| 7 | $\{11, 13, 17, 19, 23, 29, 31, 37, 41, 43,$ $47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97\}$ | |

# GARBELL D'ERATÒSTENES



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ~~1~~ | 2 | 3 | ~~4~~ | 5 | ~~6~~ | 7 | ~~8~~ | ~~9~~ | ~~10~~ |
| 11 | ~~12~~ | 13 | ~~14~~ | ~~15~~ | ~~16~~ | 17 | ~~18~~ | 19 | ~~20~~ |
| ~~21~~ | ~~22~~ | 23 | ~~24~~ | ~~25~~ | ~~26~~ | ~~27~~ | ~~28~~ | ~~29~~ | ~~30~~ |
| 31 | ~~32~~ | ~~33~~ | ~~34~~ | ~~35~~ | ~~36~~ | 37 | ~~38~~ | ~~39~~ | ~~40~~ |
| 41 | ~~42~~ | 43 | ~~44~~ | ~~45~~ | ~~46~~ | 47 | ~~48~~ | ~~49~~ | ~~50~~ |
| ~~51~~ | ~~52~~ | 53 | ~~54~~ | ~~55~~ | ~~56~~ | ~~57~~ | ~~58~~ | 59 | ~~60~~ |
| 61 | ~~62~~ | ~~63~~ | ~~64~~ | ~~65~~ | ~~66~~ | 67 | ~~68~~ | ~~69~~ | ~~70~~ |
| 71 | ~~72~~ | 73 | ~~74~~ | ~~75~~ | ~~76~~ | ~~77~~ | ~~78~~ | 79 | ~~80~~ |
| ~~81~~ | ~~82~~ | 83 | ~~84~~ | ~~85~~ | ~~86~~ | ~~87~~ | ~~88~~ | 89 | ~~90~~ |
| ~~91~~ | ~~92~~ | ~~93~~ | ~~94~~ | ~~95~~ | ~~96~~ | 97 | ~~98~~ | ~~99~~ | ~~100~~ |

- Requereix memòria però no operacions.
- S'utilitza per fer taules de primers "petits"
  El tamany de la taula emmagatzemada depèn de cada sistema.
- El primer pas d'un test de primalitat consisteix a comprovar que el candidat no és divisible per cap dels primers de la taula

# PRIMALITAT $\neq$ FACTORITZACIÓ

## Petit teorema de Fermat

Si $p$ és primer i $\gcd(p, b) = 1$, aleshores

$$b^{p-1} \equiv 1 \bmod p$$

Si trobem $b$ tal que $\gcd(n, b) = 1$ però $b^{n-1} \not\equiv 1 \bmod n$, llavors

- sabem que $n$ no és primer
- no sabem res de la factorització de $n$

$$\begin{aligned} n &= 768767544323345678 8991 \\ b &= 2 \\ \gcd(2, n) &= 1 \\ 2^{n-1} \pmod n &= 41867725323287179 42860 \neq 1 \Rightarrow n \text{ compost} \end{aligned}$$

# CERTIFICATS DE PRIMALITAT

## Wilson-Lagrange (1773)

$N$ primer $\Leftrightarrow (N-1)! \equiv -1 \pmod{N}$

Problema: si $N$ és gran, el càlcul de $(N-1)!$ és massa costós

## Lucas (1891)

Si existeix $b > 1$ tal que

- $b^{N-1} \equiv 1 \pmod{N}$
- $b^m \not\equiv 1 \pmod{N}$ per a tot $m$ divisor estricte de $N-1$,

llavors $N$ és primer.

Problema: requereix la factorització de $N-1$

# TESTS DE PRIMALITAT

## Test de Fermat

Calcular $b^{N-1} \pmod N$ i mirar si dóna 1 o no

Si $N$ és senar compost, o bé passa el test per a totes les bases, o bé la **probabilitat** que el passi per a una base arbitrària és $< 1/2$.

## Algoritme

Donat un enter $N$ senar,

- Triem aleatòriament $b$ tal que $1 < b < N$
- Calculem $d = (b, N)$ mitjançant l'algoritme d'Euclides
- Si $d > 1$, $N$ no és primer i $d$ n'és un divisor no trivial. **Fi**
- Si $d = 1$, calculem $b^{N-1} \pmod N$ mitjançant l'algoritme d'exponenciació modular
- Si el resultat és $\neq 1$, llavors $N$ no és primer. **Fi**
- Si el resultat és 1, tornem a començar

# Test de Fermat

Quan haguem fet això per a $k$ bases $b$ diferents, la probabilitat que $N$ sigui compost és menor o igual que $1/2^k$

excepte que $N$ sigui un nombre compost que passa el test per a totes les bases (nombre de Carmichael)
Per exemple, $561 = 3 \cdot 11 \cdot 17$ és un nombre de Carmichael

Problema: Hi ha infinits nombres de Carmichael i les caracteritzacions conegudes requereixen la factorització

# TEST DE MILLER-RABIN: preliminars

Si $p$ és primer, les classes mòdul $p$ formen un cos
(tot element $\neq 0$ té invers)

$$\mathbf{Z}/p\mathbf{Z} = \mathbf{F}_p = \{0, 1, \ldots, p-1\}$$

Si $p > 2$ és un nombre primer, l'equació $X^2 = 1 \bmod p$ té exactament dues solucions: $1$ i $p-1$ (és a dir, $\pm 1$)

**mod 7**: $1^2 = 1,\ 2^2 = 4,\ 3^2 = 2,\ 4^2 = 2,\ 5^2 = 4,\ 6^2 = 1$

**mod 8**: $1^2 = 1,\ 2^2 = 4,\ 3^2 = 1,\ 4^2 = 0,\ 5^2 = 1,\ 6^2 = 4,\ 7^2 = 1$

# TEST DE MILLER-RABIN

$N > 1$ senar. Sigui $N - 1 = 2^t N_0$, amb $t \geq 1$ i $N_0$ senar.
Si $b$ és un enter tal que $\gcd(b, N) = 1$, definim $x_0 = b^{N_0} \pmod N$ i els quadrats successius

$$\begin{aligned}
x_1 &= x_0^2 \pmod N && = b^{2N_0} \pmod N \\
&\dots \\
x_k &= x_{k-1}^2 \pmod N \\
&\dots \\
x_t &= x_{t-1}^2 \pmod N && = b^{2^t N_0} = b^{N-1} \pmod N.
\end{aligned}$$

Direm que $N$ passa el test de Miller-Rabin per a la base $b$ si o bé $x_0 = 1$ o bé $N - 1 \in \{x_0, x_1, \dots, x_{t-1}\}$.

La probabilitat d'error en el test de Miller-Rabin (*pseudoprimers forts*) és $\leq 1/4$

# ALGORITME: Test de Miller-Rabin amb $k$ bases

Fixem $k \geq 1$. Donat un enter senar $N$,

- Es prenen aleatòriament enters $b_1, \ldots, b_k$ tals que $1 < b_i < N$
- Per a cada $i$ es calcula $\gcd(N, b_i)$. Si algun és diferent de 1, aleshores $N$ és compost. **Fi**
- Si tots són 1, es fa el test de Miller-Rabin per a cada $b_i$. Si per a algun no el passa, $N$ és compost. **Fi**
- Si el passa per a tots, es decideix que $N$ és primer

Per a cada base fem un gcd, una exponenciació modular i menys de $t = O(\log_2 N)$ quadrats.

$$O(\ell(N)^3)$$

# ALGORITME: Test de Miller-Rabin amb $k$ bases

Fixem $k \geq 1$. Donat un enter senar $N$,

- Es prenen aleatòriament enters $b_1, \ldots, b_k$ tals que $1 < b_i < N$
- Per a cada $i$ es calcula $\gcd(N, b_i)$. Si algun és diferent de 1, aleshores $N$ és compost. **Fi**
- Si tots són 1, es fa el test de Miller-Rabin per a cada $b_i$. Si per a algun no el passa, $N$ és compost. **Fi**
- Si el passa per a tots, es decideix que $N$ és primer

Per a cada base fem un gcd, una exponenciació modular i menys de $t = O(\log_2 N)$ quadrats.

$$O(\ell(N)^3)$$

# ALGORITME: Test de Miller-Rabin amb $k$ bases

Fixem $k \geq 1$. Donat un enter senar $N$,

- Es prenen aleatòriament enters $b_1, \ldots, b_k$ tals que $1 < b_i < N$
- Per a cada $i$ es calcula $\gcd(N, b_i)$. Si algun és diferent de 1, aleshores $N$ és compost. **Fi**
- Si tots són 1, es fa el test de Miller-Rabin per a cada $b_i$. Si per a algun no el passa, $N$ és compost. **Fi**
- Si el passa per a tots, es decideix que $N$ és primer

Per a cada base fem un gcd, una exponenciació modular i menys de $t = O(\log_2 N)$ quadrats.

$$O(\ell(N)^3)$$

# ALGORITME: Test de Miller-Rabin amb $k$ bases

Fixem $k \geq 1$. Donat un enter senar $N$,

- Es prenen aleatòriament enters $b_1, \ldots, b_k$ tals que $1 < b_i < N$
- Per a cada $i$ es calcula $\gcd(N, b_i)$. Si algun és diferent de 1, aleshores $N$ és compost. **Fi**
- Si tots són 1, es fa el test de Miller-Rabin per a cada $b_i$. Si per a algun no el passa, $N$ és compost. **Fi**
- Si el passa per a tots, es decideix que $N$ és primer

Per a cada base fem un gcd, una exponenciació modular i menys de $t = O(\log_2 N)$ quadrats.

$$O(\ell(N)^3)$$

# ALGORITME: Test de Miller-Rabin amb $k$ bases

- Si el resultat és que $N$ és compost, aleshores podem estar segurs que $N$ és compost.
- Si el resultat és que $N$ és primer, hi ha una probabilitat d'error menor que $1/4^k$

(Algoritme probabilístic de Monte-Carlo)

# Generació de primers

**Entrada**: Longitud $\ell$ i paràmetre de seguretat $k$
**Sortida**: Un enter de $\ell$ bits probablement primer

1. Generar aleatòriament un enter senar $N$ de $\ell$ bits
2. Passar el test de Miller Rabin amb $k$ bases. Si respon primer, retornar $N$. Sinó, anar al pas 1.

$p_{\ell,k}$ = probabilitat que l'algoritme retorni un nombre compost

# Probabilitat d'error

I. Damgard, P. Landrock, and C. Pomerance
*Average case error estimates for the strong probable prime test*
Math. Comp. 61 (1993), 177–194

1. $p_{\ell,1} < \ell^2 4^{2-\sqrt{\ell}}$ per a $\ell \geq 2$.
2. $p_{\ell,k} < \ell^{3/2} 2^k k^{-1/2} 4^{2-\sqrt{k\ell}}$
   per a $(k = 2, \ell \geq 88)$ o $(3 \leq k \leq \ell/9,\ \ell \geq 21)$.
3. $p_{\ell,k} < \frac{7}{20}\ell\, 2^{-5k} + \frac{1}{7}\ell^{15/4} 2^{-\ell/2-2k} + 12\ell\, 2^{-\ell/4-3k}$
   per a $\ell/9 \leq k \leq \ell/4,\ \ell \geq 21$.
4. $p_{\ell,k} < \frac{1}{7}\ell^{15/4}\, 2^{-\ell/2-2k}$
   per a $k \geq \ell/4,\ \ell \geq 21$.

# Probabilitat d'error

I. Damgard, P. Landrock, and C. Pomerance
*Average case error estimates for the strong probable prime test*
Math. Comp. 61 (1993), 177–194

1. $p_{\ell,1} < \ell^2 4^{2-\sqrt{\ell}}$ per a $\ell \geq 2$.
2. $p_{\ell,k} < \ell^{3/2} 2^k k^{-1/2} 4^{2-\sqrt{k\ell}}$
   per a $(k = 2, \ell \geq 88)$ o $(3 \leq k \leq \ell/9,\ \ell \geq 21)$.
3. $p_{\ell,k} < \frac{7}{20}\ell\, 2^{-5k} + \frac{1}{7}\ell^{15/4} 2^{-\ell/2-2k} + 12\ell\, 2^{-\ell/4-3k}$
   per a $\ell/9 \leq k \leq \ell/4,\ \ell \geq 21$.
4. $p_{\ell,k} < \frac{1}{7}\ell^{15/4}\, 2^{-\ell/2-2k}$
   per a $k \geq \ell/4,\ \ell \geq 21$.

# Probabilitat d'error

I. Damgard, P. Landrock, and C. Pomerance
*Average case error estimates for the strong probable prime test*
Math. Comp. 61 (1993), 177–194

1. $p_{\ell,1} < \ell^2 4^{2-\sqrt{\ell}}$ per a $\ell \geq 2$.
2. $p_{\ell,k} < \ell^{3/2} 2^k k^{-1/2} 4^{2-\sqrt{k\ell}}$
   per a $(k = 2, \ell \geq 88)$ o $(3 \leq k \leq \ell/9,\ \ell \geq 21)$.
3. $p_{\ell,k} < \frac{7}{20}\ell\, 2^{-5k} + \frac{1}{7}\ell^{15/4} 2^{-\ell/2-2k} + 12\ell\, 2^{-\ell/4-3k}$
   per a $\ell/9 \leq k \leq \ell/4,\ \ell \geq 21$.
4. $p_{\ell,k} < \frac{1}{7}\ell^{15/4}\, 2^{-\ell/2-2k}$
   per a $k \geq \ell/4,\ \ell \geq 21$.

# Probabilitat d'error

I. Damgard, P. Landrock, and C. Pomerance
*Average case error estimates for the strong probable prime test*
Math. Comp. 61 (1993), 177–194

1. $p_{\ell,1} < \ell^2 4^{2-\sqrt{\ell}}$ per a $\ell \geq 2$.
2. $p_{\ell,k} < \ell^{3/2} 2^k k^{-1/2} 4^{2-\sqrt{k\ell}}$
   per a $(k = 2, \ell \geq 88)$ o $(3 \leq k \leq \ell/9,\ \ell \geq 21)$.
3. $p_{\ell,k} < \frac{7}{20}\ell\, 2^{-5k} + \frac{1}{7}\ell^{15/4} 2^{-\ell/2-2k} + 12\ell\, 2^{-\ell/4-3k}$
   per a $\ell/9 \leq k \leq \ell/4,\ \ell \geq 21$.
4. $p_{\ell,k} < \frac{1}{7}\ell^{15/4}\, 2^{-\ell/2-2k}$
   per a $k \geq \ell/4,\ \ell \geq 21$.

$$p_{\ell,k} \leq \frac{1}{2^i}$$

| $\ell$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ | $k=6$ | $k=7$ | $k=8$ |
|---|---|---|---|---|---|---|---|---|
| 500 | 56 | 63 | 70 | 78 | 85 | 92 | 99 | 106 |
| 550 | 65 | 72 | 79 | 86 | 93 | 100 | 107 | 113 |
| 600 | 75 | 82 | 88 | 95 | 102 | 108 | 115 | 121 |

- Per generar un primer de 1024 bits amb probabilitat d'error menor que $\frac{1}{2^{80}}$ només cal fer el test de Miller-Rabin amb 3 bases.
- Per generar un primer de 512 bits amb probabilitat d'error menor que $\frac{1}{2^{80}}$ només cal fer el test de Miller-Rabin amb 5 bases.

# Generació de primers

**Entrada**: Longitud $\ell$ i paràmetre de seguretat $k$
**Sortida**: Un enter de $\ell$ bits probablement primer

1. Generar aleatòriament un enter senar $N$ de $\ell$ bits
2. Decidir si $N$ és divisible per algun "primer petit"
3. Passar el test de Miller Rabin amb $k$ bases. Si respon primer, retornar $N$. Sinó, anar al pas 1.

Probabilitat que l'algoritme retorni un nombre compost és $\leq p_{\ell,k}$

# Generació de primers

**Entrada**: Longitud $\ell$ i paràmetre de seguretat $k$
**Sortida**: Un enter de $\ell$ bits probablement primer

1. Generar aleatòriament un enter senar $N$ de $\ell$ bits
2. Decidir si $N$ és divisible per algun "primer petit"
3. Passar el test de Miller Rabin en base 2 i $k-1$ bases aleatòries. Si respon primer, retornar $N$. Sinó, anar al pas 1.

Fixar $b = 2$ en la primera execució del test de Miller-Rabin millora el temps esperat d'execució de l'algoritme perquè l'exponenciació modular en base 2 és més eficient que en altres bases i perquè molts nombres compostos fallen el test per a aquesta base

# Generació de primers

**Entrada**: Longitud $\ell$ i paràmetre de seguretat $k$
**Sortida**: Un enter de $\ell$ bits probablement primer

1. Generar aleatòriament un enter senar $N$ de $\ell$ bits
2. Decidir si $N$ és divisible per algun "primer petit"
3. Passar el test de Miller Rabin en base 2 i $k-1$ bases aleatòries. Si respon primer, retornar $N$. Sinó, $N \leftarrow N+2$ i anar al pas 2.