Génération de nombres aléatoires et recherche de nombres premiers

Cours Turing – Semaine 9

Génération de nombres aléatoires

Premier essai: la méthode des “carrés tronqués”

0) aller chercher une graine ou seed:

ex: 19 nov à 9h31, 45 sec, 97 centième

661 = 0101000101 = un nb

1) 661 → 661^2 = 436921

6631 → 6631^2 = 43970161 → 9701

→ 9701^2 = 94109401 → 1094

Deuxième essai:
les “générateurs à congruence linéaire”

$$X \longrightarrow \underbrace{(a \cdot x + b) \pmod m}_{\in \{0, \ldots, m-1\}}$$

a, b fixés
m

ex: $(4x+3) \pmod 9$

$1 \longrightarrow 7 \longrightarrow 4 \longrightarrow 1$

$5 \longrightarrow 5$

$(4x+2) \pmod 9$

$1 \longrightarrow 6 \longrightarrow \_ \_ \_$

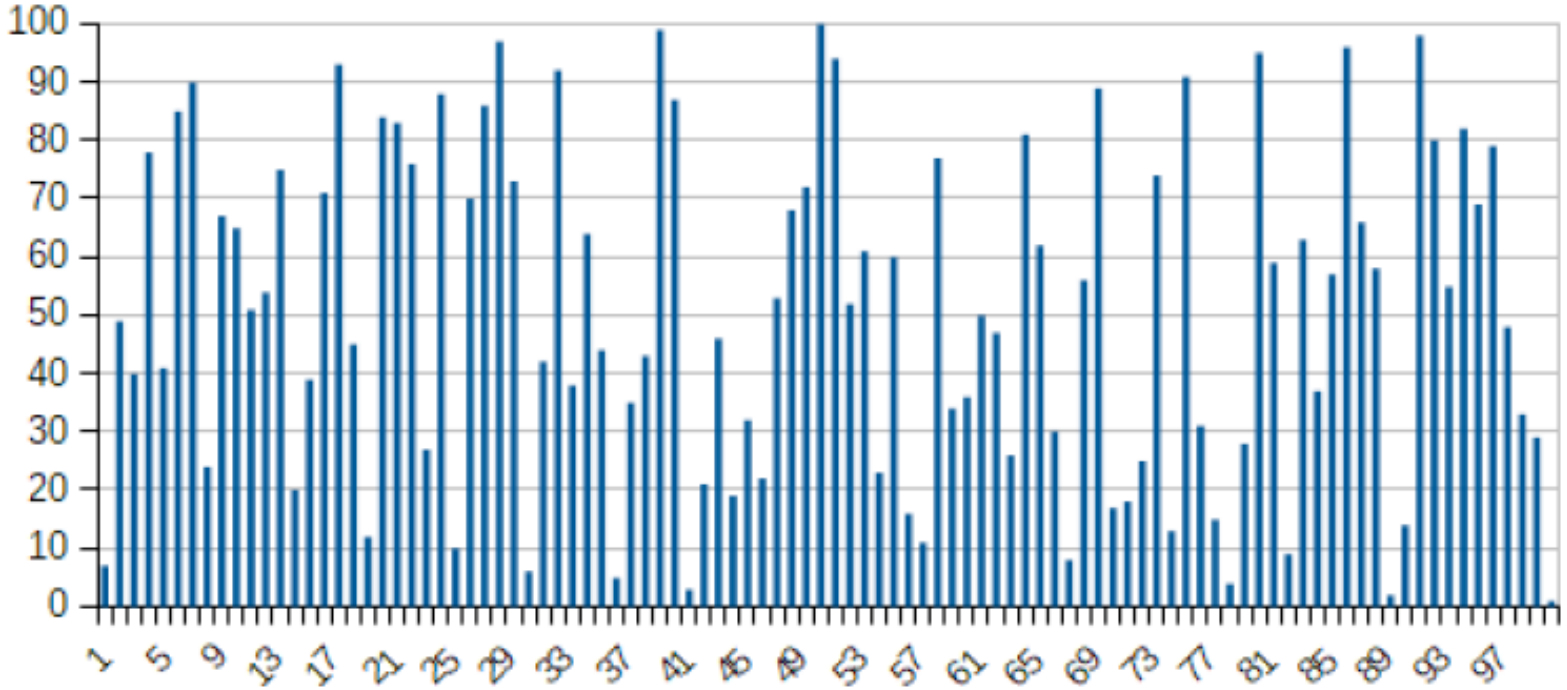
$\_ \_ \_ \_ \longrightarrow 1$

cycle de longueur max!

Générateurs à congruence linéaire

applications :

$m = 2^{32}$

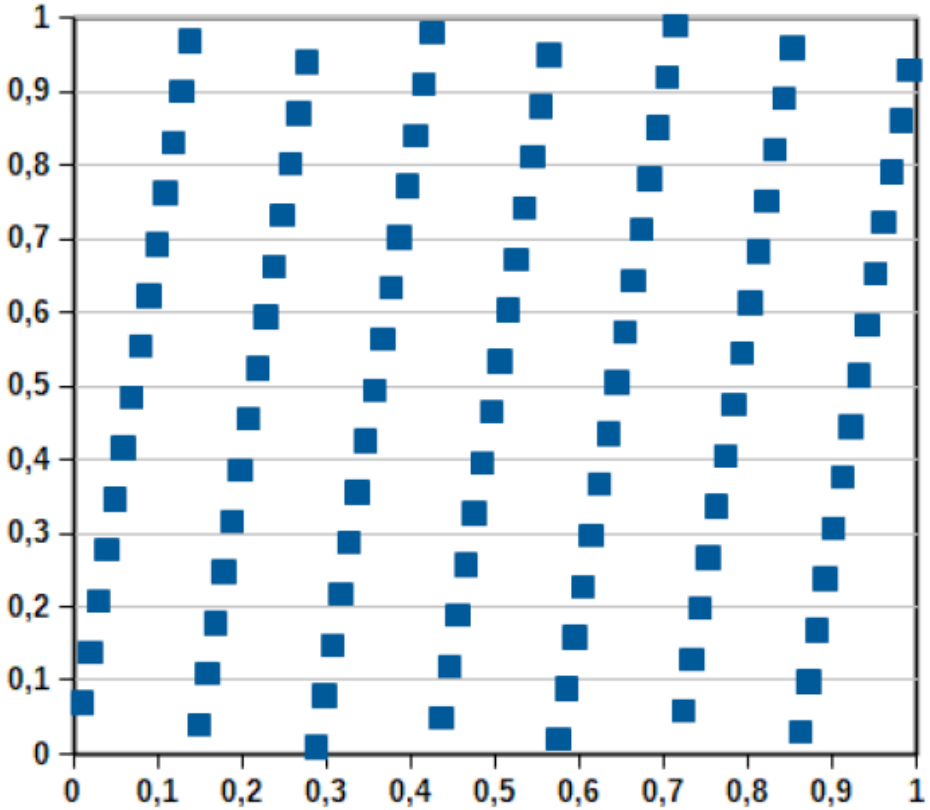


$$x \longrightarrow (7x + 0) \pmod{\underbrace{101}_{=m}}$$

Générateurs à congruence linéaire

$$\frac{(a \cdot x + b)(mod\ m)}{m}$$

$\in [0,1]$



$a = 7$
$b = 0$
$m = 101$

$\frac{x}{m} \in \{0 \ldots m-1\} \} \in [0,1]$

Opérations binaires sur des nombres entiers

$N \longrightarrow 01011101\_\_$

Ex : $3 = 0b11$

$13 = 0b1101$

Int $\longrightarrow$ bin(13) = "0b1101"

$\longrightarrow$ hex(13) = "0xd"

hex(31) = "0x1f"

XOR : $x \hat{} y = 141$

$x = 156$ $y = 17$

↓

x	10011100
y	00010001
x^y	10001101

$= 128 + 8 + 4 + 1$

ET $x \& y = 16$

$x = 156$ $y = 17$

↓

x	10011100
y	00010001
	00010000

$x \& y = 16$

Ou $x | y = 157$

$x = 156$, $y = 17$

```
  10011100
| 00010001
----------
  10011101
```

$= 157$

décalage à gauche :

$x << a$:

$x = 10011100$ 156

$x << 1$: 100111000 312

$x << 2$: 1001110000 624

$x << 3$: 10011100000 1248

décalage à droite :

$x >> a$

$x = 10011100$ 156

$x >> 1$: 1001110 0 78

$x >> 2$: 100111 00 39

$x >> 3$: 10011 000 19

Algorithme Xorshift

1. $x \longrightarrow x \hat{} (x << 13) = y$
2. $y \longrightarrow y \hat{} (y >> 17) = z$
3. $z \longrightarrow z \hat{} (z << 5) \longrightarrow x$

Exercice 1 : • que se passe-t-il si on ne fait que 1 ou 2 étapes, plutôt que 3 ?

• ⚠ séquences de nbs représentés sur 32 bits

nombres ≤ 32 bits

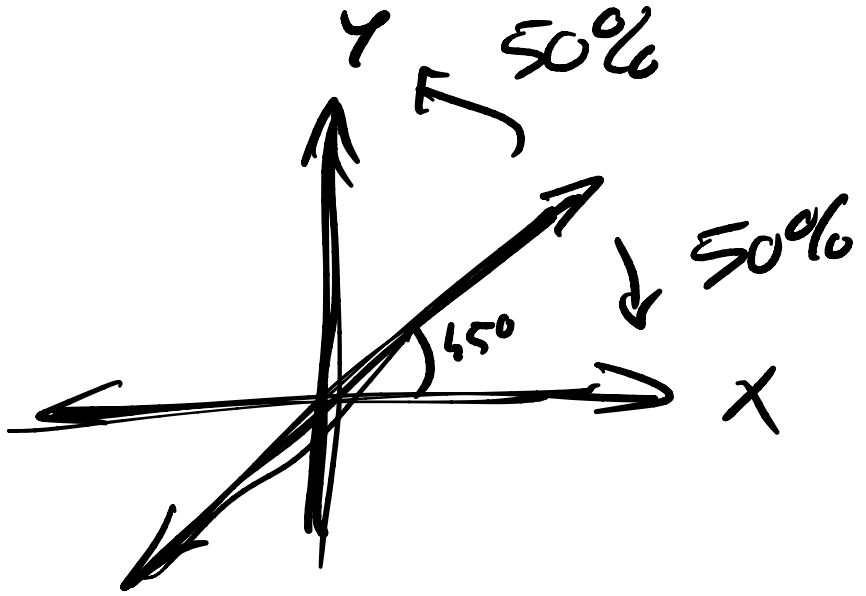
$$x \longrightarrow x \wedge (x >> 13) = y$$

$$z = y \,\&\, (0b11111111111111111111111)$$

<u>alt</u>: $z = y \,\&\, (0xffffffff)$

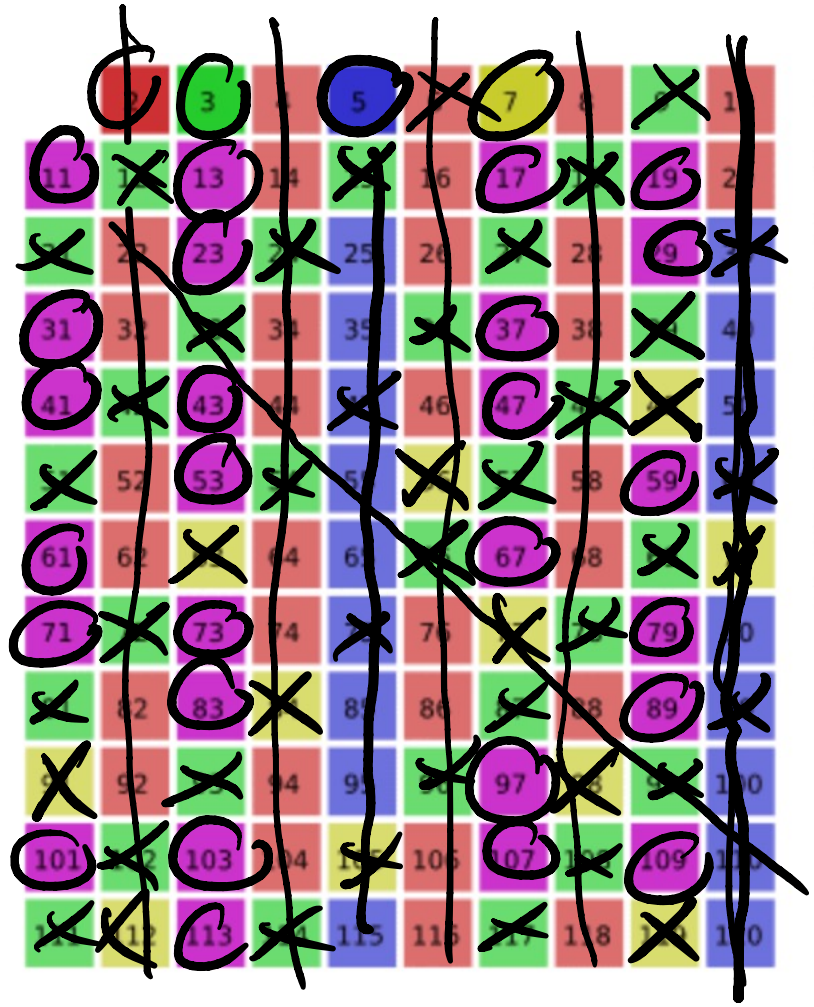
Autres méthodes

- Mersenne twister
- Théorie du chaos
- Bruit atmosphérique
- Polarisation des photons



Recherche de grands nombres premiers

Crible d’Eratosthéne



Prime numbers

2	3	5	7
11	13	17	19
23	29	31	37
41	43	47	53
59	61	67	71
73	79	83	89
97	101	103	107
109	113		

Le théorème des nombres premiers

On a besoin de nombres premiers à 100 chiffres

Un nombre à 100 chiffres est de l'ordre de 10^{101}

$N = 14573\ldots\ldots437$

100 chiffres !

parmi les nombres

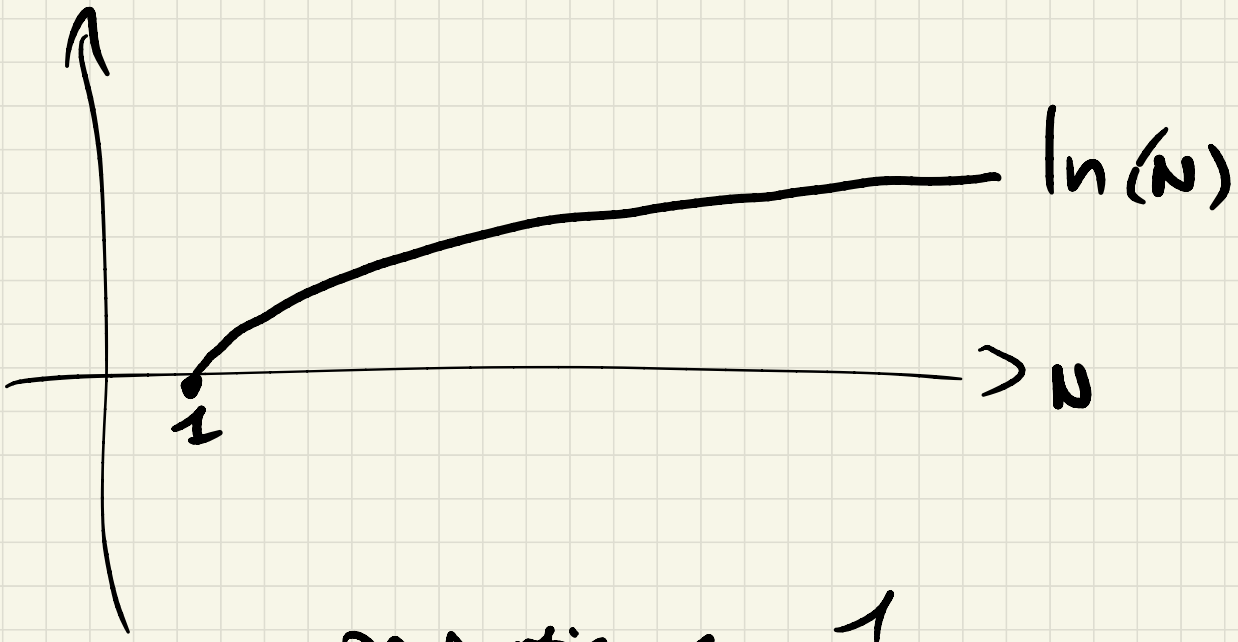
entre 1 et 10, il y a 4 nbs premiers (40%)

1 et 100, 25 (25%)

1 et 1000, 168 (17%)

donc la proportion de nbs premiers $\leq N$ décroît avec N !

Thm : cette proportion est de l'ordre de $\frac{1}{\ln(N)}$



proportion $\sim \frac{1}{\ln(N)}$ $= \frac{1}{230}$

ex: $\ln(10^{100}) = 100 \cdot \ln(10) \simeq 230$

Exercice 1

Essayez l’algorithme en n’effectuant qu’1 ou 2 étapes plutôt que 3: qu’observez-vous?

Exercice 2

Comment tester efficacement qu'un nombre est premier?

Exercice 3

Evaluer la proportion des nombres qui ne sont multiples ni de 2, ni de 3, ni de 5:

- empiriquement d’abord

- théoriquement ensuite