

# TÉLÉCOM PARIS X

## TÉLÉCODE

# SÉANCE 5 | TÉLÉCODE

## ÉLÉMENTS DE MATHÉMATIQUES

# Présenté et rédigé par Léopold Bernard

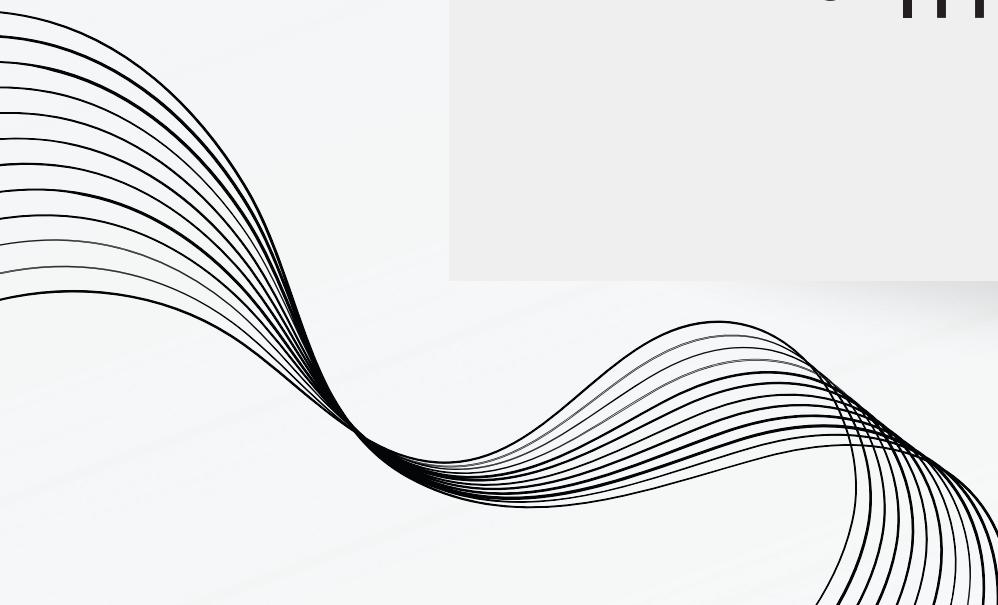
# PLAN

**01**  
**02**  
**03**  
**04**  
**05**  
**06**

- PRÉSENTATION
- EXPONENTIATION RAPIDE
- INVERSE MODULAIRE
- CRIBLE D'ERATOSTHENE
- TESTS DE PRIMALITÉ
- PROBLÈMES

# 1. PRÉSENTATION

- Algèbre
  - Maîtriser l'exponentiation rapide !
- Arithmétique
  - Crible d'Eratosthène
  - Tests de primalité : connaître le naïf
  - Test probabiliste efficace ?
  - Inverse modulaire



## 2. EXPONENTIATION RAPIDE

```
def binexp(x, n, m) :  
    ans = 1  
    while n > 0 :  
        if n % 2 :  
            ans = (ans * x) % m  
            x = (x*x) % m  
        n //= 2  
    return ans
```

# 2. EXPONENTIATION RAPIDE

- Applications
  - Puissances de matrices
  - Nombres de Fibonacci
  - Appliquer opération géométrique à un ensemble de points
  - Nombres de chemins de longueur donnée dans un graphe



# 3. INVERSE MODULAIRE

- Travaille souvent mod p (p premier)
- $p = 10^{9+7}$  est premier
- Petit théorème de Fermat :
  - a premier avec p i.e.  $\gcd(a, p) = 1$
  - Alors  $a^{p-1} \equiv 1[p]$
  - Donc l'inverse de a est  $a^{p-2}$
  - On le calcule avec exponentiation rapide !



## 4. CRIBLE D'ERATOSTHENE

- Complexité :  $O(n \log \log n)$ 
  - Quasiment linéaire !
- Comprendre visuellement
- Preuve de la complexité sur CP-algorithms
- Idée que l'on retrouve pour l'indicatrice

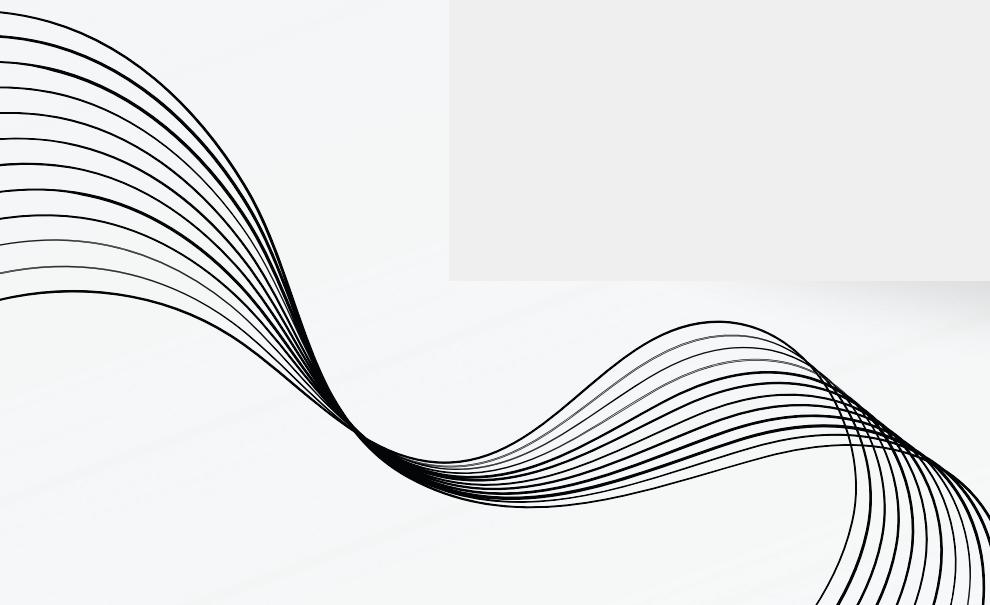


# 4. CRIBLE D'ERATOSTHENE

```
def erath(n) :  
    P = [True] * n  
    primes = []  
    P[0] = P[1] = False  
    for i in range (2, n) :  
        if P[i] :  
            primes.append(i)  
            for j in range (i*i, n, i) :  
                P[j] = False  
    return P, primes
```

# 5. TESTS DE PRIMALITÉ

- Naïf en  $O(\sqrt{n})$ 
  - chercher diviseurs jusqu'à  $\sqrt{n}$
- Probabiliste ?
  - 1<sup>e</sup> algorithme : repose sur le petit théorème de Fermat
  - Problème : nombres de Carmichael (seulement 646 inférieurs à  $10^9$ )



# 5. TESTS DE PRIMALITÉ

- Solovay-Strassen
- Résidus quadratique modulo p
  - généralisation : symbole de legendre
- Probabilité d'erreur :  $2^{\wedge}(-\text{nb\_tests})$

$T = \{a \in \mathbb{Z}/n\mathbb{Z} : \left(\frac{a}{n}\right) \equiv a^{\frac{n-1}{2}} \pmod{n}\}$  sous-groupe de  $\mathbb{Z}/n\mathbb{Z}$ , égal à ssi  $n \in \mathbb{P}$

Or s'il est strict  $|T| \leq \frac{n}{2}$  donc probabilité 1/2 qu'un nombre  $a$  tiré au hasard ne vérifie pas l'égalité



# 5. TESTS DE PRIMALITÉ

```
def jacobi(a, n) :
    # print(a, n)
    if a == 1 or n == 1:
        return 1
    elif a % 2 == 0 :
        x = -1 if (n % 8 in [3, 5]) else 1 # equiv à  $n^2 - 1 = 8 \bmod 16$ 
        return x * jacobi(a//2, n)
    else :
        # a, n impairs
        ma = (a-1) % 8
        mn = (n-1) % 8
        x = -1 if ((ma * mn) % 8 == 4) else 1
        return x * jacobi(n%a, a)
```



# 6. PROBLÈMES

CSES (mathématiques) :

Exponentiation I, II

Counting divisors

Common divisors

Sum of divisors

Binomial coefficients