# Algorithmes pour l'arithmétique II
# Cours 7

Julien Lavauzelle

Université Paris 8

Master 2 ACC – Algorithmes pour l'arithmétique

17/11/2022

# La semaine dernière

**Questions ?**

# Plan

Factorisation d'entiers

## Factorisation d'entiers

C'est un problème **ancien**, très étudié, et utile de nos jours :

- étudié par Gauss, Fermat, Mersenne, etc.
- sécurité du cryptosystème RSA, générateur Blum-Blum-Shub

## Factorisation d'entiers

C'est un problème **ancien**, très étudié, et utile de nos jours :

- étudié par Gauss, Fermat, Mersenne, etc.
- sécurité du cryptosystème RSA, générateur Blum-Blum-Shub

**Problème.** Étant donné un entier $N \geq 2$ composé, trouver sa décomposition en facteurs premiers

$$N = \prod_{i=1}^{s} p_i^{e_i} .$$

## Factorisation d'entiers

C'est un problème **ancien**, très étudié, et utile de nos jours :

- étudié par Gauss, Fermat, Mersenne, etc.
- sécurité du cryptosystème RSA, générateur Blum-Blum-Shub

**Problème.** Étant donné un entier $N \geq 2$ composé, trouver sa décomposition en facteurs premiers

$$N = \prod_{i=1}^{s} p_i^{e_i} .$$

**Sous-problème.** Étant donné $N$ composé, trouver un facteur non-trivial de $N$.

## Factorisation d'entiers

C'est un problème **ancien**, très étudié, et utile de nos jours :

- étudié par Gauss, Fermat, Mersenne, etc.
- sécurité du cryptosystème RSA, générateur Blum-Blum-Shub

**Problème.** Étant donné un entier $N \geq 2$ composé, trouver sa décomposition en facteurs premiers

$$N = \prod_{i=1}^{s} p_i^{e_i} .$$

**Sous-problème.** Étant donné $N$ composé, trouver un facteur non-trivial de $N$.

**Complexité** mesurée en fonction de la taille $\log_2(N)$ de l'entier $N$ à factoriser.

- un algorithme **polynomial** est en $O(\log_2(N)^d)$ pour un certain $d > 0$ ;
- avoir un algorithme en $N^a$ reste **exponentiel**

$$O(N^a) = O(2^{a \log_2(N)}) .$$

# Méthode des divisions successives

**Rappelons** la méthode « classique » :

DIVISIONS SUCCESSIVES POUR LA FACTORISATION D'ENTIERS

**Entrée :** un entier composé $N = \prod p_k^{e_k} \geq 2$.
**Sortie :** la factorisation de $N$ sous forme de liste $L = [(p_1, e_1), \ldots, (p_k, e_k)]$

1. $n \leftarrow N$, $m \leftarrow 2$, et $L \leftarrow [\,]$
2. **Tant que** $n \neq 1$ :
   - 2.1 **Si** $m$ divise $n$, ajouter $(m, 0)$ à $L$
   - 2.2 **Tant que** $m$ divise $n$ :
     - Remplacer $(m, i)$ par $(m, i+1)$ dans $L$
     - Diviser $n$ par $m$
   - 2.3 $m \leftarrow m + 1$
3. **Retourner** $L$.

# Méthode des divisions successives

**Rappelons** la méthode « classique » :

DIVISIONS SUCCESSIVES POUR LA FACTORISATION D'ENTIERS

**Entrée :** un entier composé $N = \prod p_k^{e_k} \geq 2$.
**Sortie :** la factorisation de $N$ sous forme de liste $L = [(p_1, e_1), \ldots, (p_k, e_k)]$

1. $n \leftarrow N$, $m \leftarrow 2$, et $L \leftarrow [\,]$
2. **Tant que** $n \neq 1$ :
   - 2.1 **Si** $m$ divise $n$, ajouter $(m, 0)$ à $L$
   - 2.2 **Tant que** $m$ divise $n$ :
     - Remplacer $(m, i)$ par $(m, i+1)$ dans $L$
     - Diviser $n$ par $m$
   - 2.3 $m \leftarrow m + 1$
3. **Retourner** $L$.

**Remarques.**

1. On peut ajouter un test de primalité efficace pour éviter de terminer la boucle pour le dernier facteur.
2. L'algorithme trouve rapidement les petits facteurs de $N$.

# Méthode des divisions successives : complexité

**Complexité.**

# Méthode des divisions successives : complexité

**Complexité.**

– Le nombre d'opérations pour la factorisation complète est en $\mathcal{O}(p_2 \log^2 N)$ où $p_2$ est le deuxième plus grand facteur premier de $N$.

# Méthode des divisions successives : complexité

**Complexité.**

- Le nombre d'opérations pour la factorisation complète est en $\mathcal{O}(p_2 \log^2 N)$ où $p_2$ est le deuxième plus grand facteur premier de $N$.
- Pour trouver le plus petit facteur $p$, complexité en $\mathcal{O}(p \log^2 N)$

# Méthode des divisions successives : complexité

**Complexité.**

- Le nombre d'opérations pour la factorisation complète est en $\mathcal{O}(p_2 \log^2 N)$ où $p_2$ est le deuxième plus grand facteur premier de $N$.
- Pour trouver le plus petit facteur $p$, complexité en $\mathcal{O}(p \log^2 N)$
- En pire cas, on a $N = p_1 p_2$ où $p_1 > p_2$ sont deux premiers proches de $\sqrt{N}$, donc une complexité en $\mathcal{O}(\sqrt{N})$.

# Méthode des divisions successives : complexité

**Complexité.**

- Le nombre d'opérations pour la factorisation complète est en $\mathcal{O}(p_2 \log^2 N)$ où $p_2$ est le deuxième plus grand facteur premier de $N$.
- Pour trouver le plus petit facteur $p$, complexité en $\mathcal{O}(p \log^2 N)$
- En pire cas, on a $N = p_1 p_2$ où $p_1 > p_2$ sont deux premiers proches de $\sqrt{N}$, donc une complexité en $\mathcal{O}(\sqrt{N})$.
- En cas moyen, on a aussi une complexité en $\mathcal{O}(\sqrt{N})$.

# Méthode des divisions successives : complexité

**Complexité.**

- Le nombre d'opérations pour la factorisation complète est en $\mathcal{O}(p_2 \log^2 N)$ où $p_2$ est le deuxième plus grand facteur premier de $N$.
- Pour trouver le plus petit facteur $p$, complexité en $\mathcal{O}(p \log^2 N)$
- En pire cas, on a $N = p_1 p_2$ où $p_1 > p_2$ sont deux premiers proches de $\sqrt{N}$, donc une complexité en $\mathcal{O}(\sqrt{N})$.
- En cas moyen, on a aussi une complexité en $\mathcal{O}(\sqrt{N})$.

Pour une **implantation simple**, en `python` :

- $<$ 1 seconde en moyenne pour des entiers aléatoires de $\sim$ 50 bits ($\sim$ 15 chiffres), plusieurs dizaines de secondes en pire cas.
- incalculable en pire cas pour des entiers de $\sim$ 100 bits.

# Méthode des divisions successives : complexité

**Complexité.**

- Le nombre d'opérations pour la factorisation complète est en $\mathcal{O}(p_2 \log^2 N)$ où $p_2$ est le deuxième plus grand facteur premier de $N$.
- Pour trouver le plus petit facteur $p$, complexité en $\mathcal{O}(p \log^2 N)$
- En pire cas, on a $N = p_1 p_2$ où $p_1 > p_2$ sont deux premiers proches de $\sqrt{N}$, donc une complexité en $\mathcal{O}(\sqrt{N})$.
- En cas moyen, on a aussi une complexité en $\mathcal{O}(\sqrt{N})$.

Pour une **implantation simple**, en `python` :

- $<$ 1 seconde en moyenne pour des entiers aléatoires de $\sim$ 50 bits ($\sim$ 15 chiffres), plusieurs dizaines de secondes en pire cas.
- incalculable en pire cas pour des entiers de $\sim$ 100 bits.

Cela pose différentes questions :

1. le « cas moyen » est-il facile ?
2. quel est le nombre moyen de facteurs à chercher ?
3. quelle est la taille des entiers à chercher ? la taille du plus gros facteur ?

# Sur les facteurs à chercher

Sur le **nombre** de facteurs à trouver :

**Fait.** Le nombre de facteurs premiers distincts d'un entier pris au hasard entre 1 et $N$ est $O(\log\log N)$.

Si $N = p_1^{e_1} \cdots p_k^{e_k}$, on a donc $\log_2 p_1 \leq \frac{\log N}{\log\log N}$ en pire cas.

# Sur les facteurs à chercher

Sur le **nombre** de facteurs à trouver :

**Fait.** Le nombre de facteurs premiers distincts d'un entier pris au hasard entre 1 et $N$ est $O(\log\log N)$.

Si $N = p_1^{e_1} \cdots p_k^{e_k}$, on a donc $\log_2 p_1 \leq \frac{\log N}{\log\log N}$ en pire cas.

Sur la **taille** des facteurs à trouver :

**Fait.** On peut démontrer qu'en moyenne $p_k \sim N^{0,62}$ et $p_{k-1} \sim N^{0,21}$. Les facteurs à chercher sont, en moyenne, de taille linéaire en celle de $N$.

# Sur les facteurs à chercher

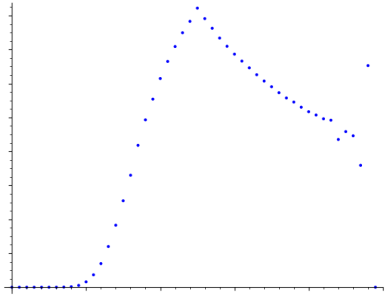
Sur le **nombre** de facteurs à trouver :

**Fait.** Le nombre de facteurs premiers distincts d'un entier pris au hasard entre 1 et $N$ est $O(\log\log N)$.

Si $N = p_1^{e_1} \cdots p_k^{e_k}$, on a donc $\log_2 p_1 \leq \frac{\log N}{\log\log N}$ en pire cas.

Sur la **taille** des facteurs à trouver :

**Fait.** On peut démontrer qu'en moyenne $p_k \sim N^{0{,}62}$ et $p_{k-1} \sim N^{0{,}21}$. Les facteurs à chercher sont, en moyenne, de taille linéaire en celle de $N$.



*Approximation expérimentale de la distribution de la taille du plus gros facteur $p_k$ de $N$* (réalisée avec plusieurs millions d'échantillons $N > 2^{40}$).

En **abscisse** : $\alpha \in [0, 1]$.
En **ordonnée** : densité de probabilité de $\log_2 p_k / \log_2 N = \alpha$.

# Aperçu des méthodes

**Aujourd'hui**, nous allons voir :

1. Algorithme de Fermat (XVIIème siècle)
2. Algorithme $\rho$ de Pollard (1975)
3. Méthode $p-1$ de Pollard (1974)
4. Aperçu rapide de ses généralisations (factorisation par groupes algébriques) : méthode $p+1$ de Williams (1982), méthode ECM de Lenstra (1987)

# Aperçu des méthodes

**Aujourd'hui**, nous allons voir :

1. Algorithme de Fermat (XVIIème siècle)
2. Algorithme $\rho$ de Pollard (1975)
3. Méthode $p-1$ de Pollard (1974)
4. Aperçu rapide de ses généralisations (factorisation par groupes algébriques) : méthode $p+1$ de Williams (1982), méthode ECM de Lenstra (1987)

Tous ces algorithmes permettent de factoriser efficacement **certains** nombres. On parle d'algorithme (à but) **spécial**.

# Aperçu des méthodes

**Aujourd'hui**, nous allons voir :

1. Algorithme de Fermat (XVIIème siècle)
2. Algorithme $\rho$ de Pollard (1975)
3. Méthode $p-1$ de Pollard (1974)
4. Aperçu rapide de ses généralisations (factorisation par groupes algébriques) : méthode $p+1$ de Williams (1982), méthode ECM de Lenstra (1987)

Tous ces algorithmes permettent de factoriser efficacement **certains** nombres. On parle d'algorithme (à but) **spécial**.

À la prochaine séance, nous verrons des méthodes **génériques**, qui deviennent intéressantes pour de grandes valeurs de $N$.

# Plan

# Algorithme de Fermat : idée

On suppose que $N$ est impair (sinon on divise par 2).

# Algorithme de Fermat : idée

On suppose que $N$ est impair (sinon on divise par 2).

**Motivation.** Si l'on sait que $N$ admet deux facteurs « proches », peut-on en tirer parti ?

# Algorithme de Fermat : idée

On suppose que $N$ est impair (sinon on divise par 2).

**Motivation.** Si l'on sait que $N$ admet deux facteurs « proches », peut-on en tirer parti ?

**Remarque fondamentale.** Si $N = pq$, alors

$$N = (t+s)(t-s) = t^2 - s^2 \quad \text{où} \quad t = \frac{p+q}{2} \text{ et } s = \frac{p-q}{2}.$$

# Algorithme de Fermat : idée

On suppose que $N$ est impair (sinon on divise par 2).

**Motivation.** Si l'on sait que $N$ admet deux facteurs « proches », peut-on en tirer parti ?

**Remarque fondamentale.** Si $N = pq$, alors

$$N = (t+s)(t-s) = t^2 - s^2 \quad \text{où} \quad t = \frac{p+q}{2} \text{ et } s = \frac{p-q}{2}.$$

**Idée :** En partant de $t = \lceil \sqrt{N} \rceil$, on teste si $c \coloneqq t^2 - N$ s'écrit comme un carré $s^2$.
Si c'est le cas, on a trouvé $p = t + s$ et $q = t - s$.

# Algorithme de Fermat : idée

On suppose que $N$ est impair (sinon on divise par 2).

**Motivation.** Si l'on sait que $N$ admet deux facteurs « proches », peut-on en tirer parti ?

**Remarque fondamentale.** Si $N = pq$, alors

$$N = (t+s)(t-s) = t^2 - s^2 \quad \text{où} \quad t = \frac{p+q}{2} \text{ et } s = \frac{p-q}{2}.$$

**Idée :** En partant de $t = \lceil \sqrt{N} \rceil$, on teste si $c \coloneqq t^2 - N$ s'écrit comme un carré $s^2$.
Si c'est le cas, on a trouvé $p = t + s$ et $q = t - s$.

ALGORITHME DE FERMAT POUR LA FACTORISATION D'ENTIERS

**Entrée :** un entier composé $N \geq 3$.
**Sortie :** un facteur propre de $N$.

1. Calculer $t = \lceil \sqrt{N} \rceil$ et $c = t^2 - N$.
2. **Tant que** $c$ n'est pas un carré :
   - incrémenter $t \leftarrow t + 1$
   - calculer $c \leftarrow c + 2t + 1$
3. Calculer une racine carrée $s$ de $c$.
4. **Retourner** $t + s$ et $t - s$.

# Algorithme de Fermat : complexité

ALGORITHME DE FERMAT POUR LA FACTORISATION D'ENTIERS

**Entrée :** un entier composé $N \geq 2$.
**Sortie :** un facteur propre de $N$.

1. Calculer $t = \lceil \sqrt{N} \rceil$ et $c = t^2 - N$.
2. **Tant que** $c$ n'est pas un carré :
   - incrémenter $t \leftarrow t + 1$
   - calculer $c \leftarrow c + 2t + 1$
3. Calculer une racine carrée $s$ de $c$.
4. **Retourner** $t + s$ et $t - s$.

# Algorithme de Fermat : complexité

ALGORITHME DE FERMAT POUR LA FACTORISATION D'ENTIERS

**Entrée :** un entier composé $N \geq 2$.
**Sortie :** un facteur propre de $N$.

1. Calculer $t = \lceil \sqrt{N} \rceil$ et $c = t^2 - N$.
2. **Tant que** $c$ n'est pas un carré :
   - incrémenter $t \leftarrow t + 1$
   - calculer $c \leftarrow c + 2t + 1$
3. Calculer une racine carrée $s$ de $c$.
4. **Retourner** $t + s$ et $t - s$.

**Complexité :**

# Algorithme de Fermat : complexité

ALGORITHME DE FERMAT POUR LA FACTORISATION D'ENTIERS

**Entrée :** un entier composé $N \geq 2$.
**Sortie :** un facteur propre de $N$.

1. Calculer $t = \lceil \sqrt{N} \rceil$ et $c = t^2 - N$.
2. **Tant que** $c$ n'est pas un carré :
   - incrémenter $t \leftarrow t + 1$
   - calculer $c \leftarrow c + 2t + 1$
3. Calculer une racine carrée $s$ de $c$.
4. **Retourner** $t + s$ et $t - s$.

**Complexité :**

- Le pire cas pour l'algorithme de Fermat est pour $N = 3p$ avec $p$ premier. On alors $t = \frac{p+3}{2} = \frac{N+9}{6}$, donc $O(N)$ étapes.

# Algorithme de Fermat : complexité

ALGORITHME DE FERMAT POUR LA FACTORISATION D'ENTIERS

**Entrée :** un entier composé $N \geq 2$.
**Sortie :** un facteur propre de $N$.

1. Calculer $t = \lceil \sqrt{N} \rceil$ et $c = t^2 - N$.
2. **Tant que** $c$ n'est pas un carré :
   - incrémenter $t \leftarrow t + 1$
   - calculer $c \leftarrow c + 2t + 1$
3. Calculer une racine carrée $s$ de $c$.
4. **Retourner** $t + s$ et $t - s$.

**Complexité :**

- Le pire cas pour l'algorithme de Fermat est pour $N = 3p$ avec $p$ premier. On alors $t = \frac{p+3}{2} = \frac{N+9}{6}$, donc $O(N)$ étapes.
- S'il existe un facteur $d$ de $N$ tel que $|d - \sqrt{N}| \leq N^{1/4}$, alors l'algorithme de Fermat donne le résultat en 1 étape !

# Algorithme de Fermat : complexité

ALGORITHME DE FERMAT POUR LA FACTORISATION D'ENTIERS

**Entrée :** un entier composé $N \geq 2$.
**Sortie :** un facteur propre de $N$.

1. Calculer $t = \lceil \sqrt{N} \rceil$ et $c = t^2 - N$.
2. **Tant que** $c$ n'est pas un carré :
   - incrémenter $t \leftarrow t + 1$
   - calculer $c \leftarrow c + 2t + 1$
3. Calculer une racine carrée $s$ de $c$.
4. **Retourner** $t + s$ et $t - s$.

**Complexité :**

- Le pire cas pour l'algorithme de Fermat est pour $N = 3p$ avec $p$ premier. On alors $t = \frac{p+3}{2} = \frac{N+9}{6}$, donc $O(N)$ étapes.
- S'il existe un facteur $d$ de $N$ tel que $|d - \sqrt{N}| \leq N^{1/4}$, alors l'algorithme de Fermat donne le résultat en 1 étape !
- La complexité en fonction de $p$ où $N = pq$, et $p > q$, est $O(\frac{(p-\sqrt{N})^2}{2p})$

ALGORITHME DE FERMAT POUR LA FACTORISATION D'ENTIERS

**Entrée :** un entier composé $N \geq 2$.
**Sortie :** un facteur propre de $N$.

1. Calculer $t = \lceil \sqrt{N} \rceil$ et $c = t^2 - N$.
2. **Tant que** $c$ n'est pas un carré :
   - incrémenter $t \leftarrow t + 1$
   - calculer $c \leftarrow c + 2t + 1$
3. Calculer une racine carrée $s$ de $c$.
4. **Retourner** $t + s$ et $t - s$.

**Complexité :**

- Le pire cas pour l'algorithme de Fermat est pour $N = 3p$ avec $p$ premier. On alors $t = \frac{p+3}{2} = \frac{N+9}{6}$, donc $O(N)$ étapes.
- S'il existe un facteur $d$ de $N$ tel que $|d - \sqrt{N}| \leq N^{1/4}$, alors l'algorithme de Fermat donne le résultat en 1 étape !
- La complexité en fonction de $p$ où $N = pq$, et $p > q$, est $O(\frac{(p-\sqrt{N})^2}{2p})$
- En cas moyen, le nombre d'itérations est $O(N^{2/3})$.

# Algorithme de Fermat : remarques

ALGORITHME DE FERMAT POUR LA FACTORISATION D'ENTIERS

**Entrée :** un entier composé $N \geq 2$.
**Sortie :** un facteur propre de $N$.

1. Calculer $t = \lceil \sqrt{N} \rceil$ et $c = t^2 - N$.
2. **Tant que** $c$ n'est pas un carré :
   - incrémenter $t \leftarrow t + 1$
   - calculer $c \leftarrow c + 2t + 1$
3. Calculer une racine carrée $s$ de $c$.
4. **Retourner** $t + s$ et $t - s$.

# Algorithme de Fermat : remarques

ALGORITHME DE FERMAT POUR LA FACTORISATION D'ENTIERS

**Entrée :** un entier composé $N \geq 2$.
**Sortie :** un facteur propre de $N$.

1. Calculer $t = \lceil \sqrt{N} \rceil$ et $c = t^2 - N$.
2. **Tant que** $c$ n'est pas un carré :
   - incrémenter $t \leftarrow t + 1$
   - calculer $c \leftarrow c + 2t + 1$
3. Calculer une racine carrée $s$ de $c$.
4. **Retourner** $t + s$ et $t - s$.

**Remarques :**

# Algorithme de Fermat : remarques

ALGORITHME DE FERMAT POUR LA FACTORISATION D'ENTIERS

**Entrée :** un entier composé $N \geq 2$.
**Sortie :** un facteur propre de $N$.

1. Calculer $t = \lceil \sqrt{N} \rceil$ et $c = t^2 - N$.
2. **Tant que** $c$ n'est pas un carré :
   - incrémenter $t \leftarrow t + 1$
   - calculer $c \leftarrow c + 2t + 1$
3. Calculer une racine carrée $s$ de $c$.
4. **Retourner** $t + s$ et $t - s$.

**Remarques :**

- On peut **dissocier** le test « $c$ n'est pas un carré » et l'extraction de sa racine carrée. Par exemple, pour éliminer rapidement les non-carrés, on peut tester la résiduosité quadratique de $c$ modulo de petits premiers.

# Algorithme de Fermat : remarques

ALGORITHME DE FERMAT POUR LA FACTORISATION D'ENTIERS

**Entrée :** un entier composé $N \geq 2$.
**Sortie :** un facteur propre de $N$.

1. Calculer $t = \lceil \sqrt{N} \rceil$ et $c = t^2 - N$.
2. **Tant que** $c$ n'est pas un carré :
   - incrémenter $t \leftarrow t + 1$
   - calculer $c \leftarrow c + 2t + 1$
3. Calculer une racine carrée $s$ de $c$.
4. **Retourner** $t + s$ et $t - s$.

**Remarques :**

- On peut **dissocier** le test « $c$ n'est pas un carré » et l'extraction de sa racine carrée. Par exemple, pour éliminer rapidement les non-carrés, on peut tester la résiduosité quadratique de $c$ modulo de petits premiers.
- Pour éviter des cas défavorables, on peut également multiplier $N$ par des petits entiers aléatoires (voir exemple suivant). En **combinant avec les divisions successives**, on peut alors obtenir une complexité en $O(N^{1/3})$

# Algorithme de Fermat : exemples

**Exemple 1 :** $N = 2183$.

# Algorithme de Fermat : exemples

**Exemple 1 :** $N = 2183$.

On a $\lceil \sqrt{N} \rceil = 47$.

# Algorithme de Fermat : exemples

**Exemple 1 :** $N = 2183$.

On a $\lceil \sqrt{N} \rceil = 47$.

# Algorithme de Fermat : exemples

**Exemple 1 :** $N = 2183$.

On a $\lceil \sqrt{N} \rceil = 47$.

| $t$ | $t^2 - N$ | carré ? |
|---|---|---|

# Algorithme de Fermat : exemples

**Exemple 1 :** $N = 2183$.

On a $\lceil \sqrt{N} \rceil = 47$.

| $t$ | $t^2 - N$ | carré ? |
|---|---|---|
| 47 | 26 | $\times$ |

# Algorithme de Fermat : exemples

**Exemple 1 :** $N = 2183$.

On a $\lceil \sqrt{N} \rceil = 47$.

| $t$ | $t^2 - N$ | carré ? |
|---|---|---|
| 47 | 26 | $\times$ |
| 48 | 121 | ✓ |

# Algorithme de Fermat : exemples

**Exemple 1 :** $N = 2183$.

On a $\lceil \sqrt{N} \rceil = 47$.

| $t$ | $t^2 - N$ | carré ? |
|---|---|---|
| 47 | 26 | $\times$ |
| 48 | 121 | ✓ |

Donc $N = (48 - 11)(48 + 11) = 59 \times 37$

# Algorithme de Fermat : exemples

**Exemple 1 :** $N = 2183$.

On a $\lceil \sqrt{N} \rceil = 47$.

| $t$ | $t^2 - N$ | carré ? |
|---|---|---|
| 47 | 26 | $\times$ |
| 48 | 121 | ✓ |

Donc $N = (48 - 11)(48 + 11) = 59 \times 37$

**Exemple 2.** $N = 2581$.

On a $\lceil \sqrt{N} \rceil = 51$.

| $t$ | $t^2 - N$ | carré ? |
|---|---|---|
| 51 | 20 | $\times$ |
| 52 | 123 | $\times$ |
| 53 | 228 | $\times$ |
| 54 | 335 | $\times$ |
| 55 | 444 | $\times$ |
| 56 | 555 | $\times$ |
| 57 | 668 | $\times$ |
| 58 | 783 | $\times$ |
| 59 | 900 | ✓ |

Donc $N = (59 - 30)(59 + 30) = 29 \times 89$.

# Algorithme de Fermat : exemples

**Exemple 1 :** $N = 2183$.

On a $\lceil \sqrt{N} \rceil = 47$.

| $t$ | $t^2 - N$ | carré ? |
|---|---|---|
| 47 | 26 | × |
| 48 | 121 | ✓ |

Donc $N = (48 - 11)(48 + 11) = 59 \times 37$

**Exemple 2.** $N = 2581$.

On a $\lceil \sqrt{N} \rceil = 51$.

| $t$ | $t^2 - N$ | carré ? |
|---|---|---|
| 51 | 20 | × |
| 52 | 123 | × |
| 53 | 228 | × |
| 54 | 335 | × |
| 55 | 444 | × |
| 56 | 555 | × |
| 57 | 668 | × |
| 58 | 783 | × |
| 59 | 900 | ✓ |

Donc $N = (59 - 30)(59 + 30) = 29 \times 89$.

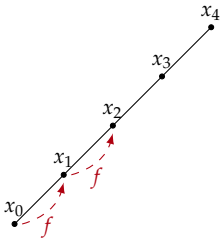
**Remarque.** En reprenant l'**Exemple 2** et en multipliant $N$ par le petit nombre premier $p = 3$, on aurait obtenu le résultat en une seule étape ! En effet, $pN = 89 \times 87$.

# Plan

# Périodicité des suites récurrentes finies

**Contexte.** Soit $E$ un ensemble fini de cardinal $M$, et $f$ une application $E \to E$. Étant donné $x_0 \in E$, on définit une suite récurrente $(x_t)_{t\geq 0}$ par

$$x_{t+1} = f(x_t)\,.$$



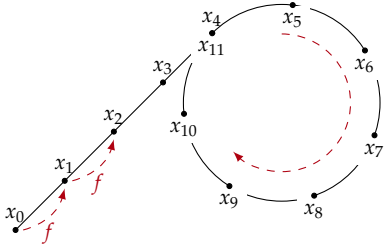
# Périodicité des suites récurrentes finies

**Contexte.** Soit $E$ un ensemble fini de cardinal $M$, et $f$ une application $E \to E$. Étant donné $x_0 \in E$, on définit une suite récurrente $(x_t)_{t\geq 0}$ par

$$x_{t+1} = f(x_t)\,.$$

Alors on sait que $(x_t)_{t\geq 0}$ est **ultimement périodique** :

$$\exists T \geq 0, \exists \tau \geq 1, \forall t \geq T,\ x_{t+\tau} = x_t\,.$$



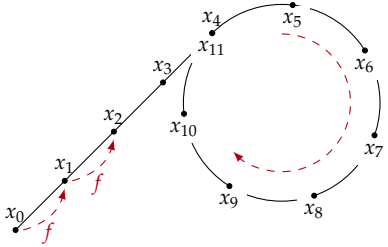
# Périodicité des suites récurrentes finies

**Contexte.** Soit $E$ un ensemble fini de cardinal $M$, et $f$ une application $E \to E$. Étant donné $x_0 \in E$, on définit une suite récurrente $(x_t)_{t\geq 0}$ par

$$x_{t+1} = f(x_t)\,.$$

Alors on sait que $(x_t)_{t\geq 0}$ est **ultimement périodique** :

$$\exists T \geq 0, \exists \tau \geq 1, \forall t \geq T,\ x_{t+\tau} = x_t\,.$$




**Définitions.**

- Le plus petit $\tau$ est appelé la **période**.
- Le plus petit $T$ est appelé la **prépériode**.

# Périodicité des suites récurrentes finies

**Contexte.** Soit $E$ un ensemble fini de cardinal $M$, et $f$ une application $E \to E$. Étant donné $x_0 \in E$, on définit une suite récurrente $(x_t)_{t\geq 0}$ par

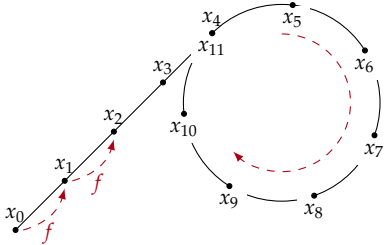
$$x_{t+1} = f(x_t)\,.$$

Alors on sait que $(x_t)_{t\geq 0}$ est **ultimement périodique** :

$$\exists T \geq 0, \exists \tau \geq 1, \forall t \geq T,\ x_{t+\tau} = x_t\,.$$

**Définitions.**

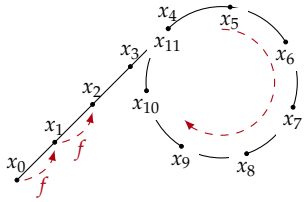
- Le plus petit $\tau$ est appelé la **période**.
- Le plus petit $T$ est appelé la **prépériode**.




Sur l'exemple, on a

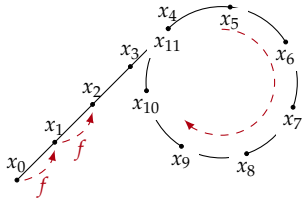
- une période de 7,
- et une prépériode de 11.

# Périodicité des suites récurrentes finies




**Question.** Si $f$ et $x_0$ sont aléatoires, quelle est la taille typique de $T$ ?

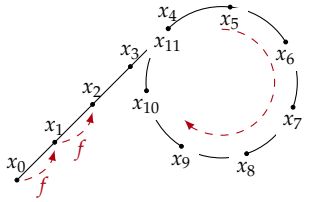
# Périodicité des suites récurrentes finies




**Question.** Si $f$ et $x_0$ sont aléatoires, quelle est la taille typique de $T$ ?

**Observation.** On a $T > m$, si et seulement si les $\{x_t\}_{t \leq m}$ sont distincts.

# Périodicité des suites récurrentes finies



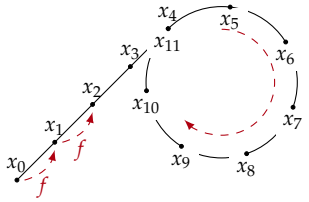

**Question.** Si $f$ et $x_0$ sont aléatoires, quelle est la taille typique de $T$ ?

**Observation.** On a $T > m$, si et seulement si les $\{x_t\}_{t \le m}$ sont distincts.

Si $f$ donne une suite aléatoire, on a donc d'après le **paradoxe des anniversaires** :

$$\begin{aligned}\mathbb{P}(T > m) &= \mathbb{P}(x_1 \neq x_0) \times \mathbb{P}(x_2 \notin \{x_1, x_0\}) \times \cdots \times \mathbb{P}(x_m \notin \{x_{m-1}, \ldots, x_0\}) \\ &= \left(1 - \frac{1}{M}\right)\left(1 - \frac{2}{M}\right) \ldots \left(1 - \frac{m}{M}\right) \simeq e^{-m^2/2M} \quad \text{pour } m \ll M\,.\end{aligned}$$

# Périodicité des suites récurrentes finies




**Question.** Si $f$ et $x_0$ sont aléatoires, quelle est la taille typique de $T$ ?

**Observation.** On a $T > m$, si et seulement si les $\{x_t\}_{t \le m}$ sont distincts.
Si $f$ donne une suite aléatoire, on a donc d'après le **paradoxe des anniversaires** :

$$\begin{aligned}\mathbb{P}(T > m) &= \mathbb{P}(x_1 \neq x_0) \times \mathbb{P}(x_2 \notin \{x_1, x_0\}) \times \cdots \times \mathbb{P}(x_m \notin \{x_{m-1}, \ldots, x_0\}) \\ &= \left(1 - \frac{1}{M}\right)\left(1 - \frac{2}{M}\right) \ldots \left(1 - \frac{m}{M}\right) \simeq e^{-m^2/2M} \quad \text{pour } m \ll M\,.\end{aligned}$$

**Conséquence.** Avec grande probabilité, la suite $(x_t)_{t \ge 0}$ a une prépériode de taille $O(\sqrt{M})$.

# Application à la factorisation

Soit $N \in \mathbb{Z}$ à factoriser, et $p$ son plus petit facteur premier, inconnu.

**Idée :** On prend $E = \mathbb{Z}/N\mathbb{Z}$, et on choisit $x_0 \in \mathbb{Z}/NZ$ aléatoire et $f \in \mathbb{Z}/N\mathbb{Z}[X]$ au comportement itéré aléatoire (hypothèse heuristique). Alors,

- la suite $(x_t \mod p)_{t \geq 0}$ a une prépériode de taille $O(\sqrt{p})$,
- sans connaître $p$, on peut détecter une collision pour $x_t \mod p$ en calculant $\mathrm{pgcd}(x_j - x_i, N)$.

# Application à la factorisation

Soit $N \in \mathbb{Z}$ à factoriser, et $p$ son plus petit facteur premier, inconnu.

**Idée :** On prend $E = \mathbb{Z}/N\mathbb{Z}$, et on choisit $x_0 \in \mathbb{Z}/NZ$ aléatoire et $f \in \mathbb{Z}/N\mathbb{Z}[X]$ au comportement itéré aléatoire (hypothèse heuristique). Alors,

- la suite $(x_t \mod p)_{t \geq 0}$ a une prépériode de taille $O(\sqrt{p})$,
- sans connaître $p$, on peut détecter une collision pour $x_t \mod p$ en calculant $\mathrm{pgcd}(x_j - x_i, N)$.

En **pratique**, des fonctions du type $f(z) = z^2 + a$, avec $a \neq 0$ (ex : $a = 1$), ont un comportement suffisamment aléatoire pour que l'analyse probabiliste tienne.

**Leur avantage :** très efficace à calculer.

# Application à la factorisation

Soit $N \in \mathbb{Z}$ à factoriser, et $p$ son plus petit facteur premier, inconnu.

**Idée :** On prend $E = \mathbb{Z}/N\mathbb{Z}$, et on choisit $x_0 \in \mathbb{Z}/NZ$ aléatoire et $f \in \mathbb{Z}/N\mathbb{Z}[X]$ au comportement itéré aléatoire (hypothèse heuristique). Alors,

- la suite $(x_t \mod p)_{t \geq 0}$ a une prépériode de taille $O(\sqrt{p})$,
- sans connaître $p$, on peut détecter une collision pour $x_t \mod p$ en calculant $\mathrm{pgcd}(x_j - x_i, N)$.

En **pratique**, des fonctions du type $f(z) = z^2 + a$, avec $a \neq 0$ (ex : $a = 1$), ont un comportement suffisamment aléatoire pour que l'analyse probabiliste tienne.

**Leur avantage :** très efficace à calculer.

**Question.** Comment détecter la collision $\mathrm{pgcd}(x_j - x_i, N)$ ?

# Application à la factorisation

Soit $N \in \mathbb{Z}$ à factoriser, et $p$ son plus petit facteur premier, inconnu.

**Idée :** On prend $E = \mathbb{Z}/N\mathbb{Z}$, et on choisit $x_0 \in \mathbb{Z}/NZ$ aléatoire et $f \in \mathbb{Z}/N\mathbb{Z}[X]$ au comportement itéré aléatoire (hypothèse heuristique). Alors,

- la suite $(x_t \mod p)_{t \geq 0}$ a une prépériode de taille $O(\sqrt{p})$,
- sans connaître $p$, on peut détecter une collision pour $x_t \mod p$ en calculant $\mathrm{pgcd}(x_j - x_i, N)$.

En **pratique**, des fonctions du type $f(z) = z^2 + a$, avec $a \neq 0$ (ex : $a = 1$), ont un comportement suffisamment aléatoire pour que l'analyse probabiliste tienne.

**Leur avantage :** très efficace à calculer.

**Question.** Comment détecter la collision $\mathrm{pgcd}(x_j - x_i, N)$ ?

**Méthode naïve :**

- pour chaque nouvel $x_j$ calculé, on teste la collision potentielle avec les $x_i$ calculés antérieurement ;

# Application à la factorisation

Soit $N \in \mathbb{Z}$ à factoriser, et $p$ son plus petit facteur premier, inconnu.

**Idée :** On prend $E = \mathbb{Z}/N\mathbb{Z}$, et on choisit $x_0 \in \mathbb{Z}/NZ$ aléatoire et $f \in \mathbb{Z}/N\mathbb{Z}[X]$ au comportement itéré aléatoire (hypothèse heuristique). Alors,

- la suite $(x_t \mod p)_{t \geq 0}$ a une prépériode de taille $O(\sqrt{p})$,
- sans connaître $p$, on peut détecter une collision pour $x_t \mod p$ en calculant $\mathrm{pgcd}(x_j - x_i, N)$.

En **pratique**, des fonctions du type $f(z) = z^2 + a$, avec $a \neq 0$ (ex : $a = 1$), ont un comportement suffisamment aléatoire pour que l'analyse probabiliste tienne.

**Leur avantage :** très efficace à calculer.

**Question.** Comment détecter la collision $\mathrm{pgcd}(x_j - x_i, N)$ ?

**Méthode naïve :**

- pour chaque nouvel $x_j$ calculé, on teste la collision potentielle avec les $x_i$ calculés antérieurement ;
- complexité quadratique en $j$ en temps, linéaire en espace.

# Application à la factorisation

Soit $N \in \mathbb{Z}$ à factoriser, et $p$ son plus petit facteur premier, inconnu.

**Idée :** On prend $E = \mathbb{Z}/N\mathbb{Z}$, et on choisit $x_0 \in \mathbb{Z}/NZ$ aléatoire et $f \in \mathbb{Z}/N\mathbb{Z}[X]$ au comportement itéré aléatoire (hypothèse heuristique). Alors,

- la suite $(x_t \mod p)_{t \geq 0}$ a une prépériode de taille $O(\sqrt{p})$,
- sans connaître $p$, on peut détecter une collision pour $x_t \mod p$ en calculant $\mathrm{pgcd}(x_j - x_i, N)$.

En **pratique**, des fonctions du type $f(z) = z^2 + a$, avec $a \neq 0$ (ex : $a = 1$), ont un comportement suffisamment aléatoire pour que l'analyse probabiliste tienne.

**Leur avantage :** très efficace à calculer.

**Question.** Comment détecter la collision $\mathrm{pgcd}(x_j - x_i, N)$ ?

**Méthode naïve :**

- pour chaque nouvel $x_j$ calculé, on teste la collision potentielle avec les $x_i$ calculés antérieurement ;
- complexité quadratique en $j$ en temps, linéaire en espace.

**Remarque :** si $\mathrm{pgcd}(x_j - x_i, N) = N$ sans avoir obtenu de facteur propre auparavant, il faut changer $x_0$.

# Application à la factorisation

Soit $N \in \mathbb{Z}$ à factoriser, et $p$ son plus petit facteur premier, inconnu.

**Idée :** On prend $E = \mathbb{Z}/N\mathbb{Z}$, et on choisit $x_0 \in \mathbb{Z}/NZ$ aléatoire et $f \in \mathbb{Z}/N\mathbb{Z}[X]$ au comportement itéré aléatoire (hypothèse heuristique). Alors,

- la suite $(x_t \mod p)_{t \geq 0}$ a une prépériode de taille $O(\sqrt{p})$,
- sans connaître $p$, on peut détecter une collision pour $x_t \mod p$ en calculant $\mathrm{pgcd}(x_j - x_i, N)$.

En **pratique**, des fonctions du type $f(z) = z^2 + a$, avec $a \neq 0$ (ex : $a = 1$), ont un comportement suffisamment aléatoire pour que l'analyse probabiliste tienne.

**Leur avantage :** très efficace à calculer.

**Question.** Comment détecter la collision $\mathrm{pgcd}(x_j - x_i, N)$ ?

**Méthode naïve :**

- pour chaque nouvel $x_j$ calculé, on teste la collision potentielle avec les $x_i$ calculés antérieurement ;
- complexité quadratique en $j$ en temps, linéaire en espace.

**Remarque :** si $\mathrm{pgcd}(x_j - x_i, N) = N$ sans avoir obtenu de facteur propre auparavant, il faut changer $x_0$.

$\Longrightarrow$ Il existe un meilleur algorithme, dû à Floyd.

# Algorithme de Floyd (années 1960)

**Lemme.** Soit $(x_t)_{t\geq 0}$ une suite de prépériode $T$ et de période $\tau$. Alors, il existe $i \leq T-1$ tel que $x_{2i} = x_i$.

# Algorithme de Floyd (années 1960)

**Lemme.** Soit $(x_t)_{t\geq 0}$ une suite de prépériode $T$ et de période $\tau$. Alors, il existe $i \leq T-1$ tel que $x_{2i} = x_i$.

**Preuve.** La différence entre les indices $2i$ et $i$ vaut $i$. Au moins l'un des $i \leq T-1$ est divisible par la période $\tau$, ce qui assure que $x_{2i} = x_i$.

# Algorithme de Floyd (années 1960)

**Lemme.** Soit $(x_t)_{t\geq 0}$ une suite de prépériode $T$ et de période $\tau$. Alors, il existe $i \leq T-1$ tel que $x_{2i} = x_i$.

**Preuve.** La différence entre les indices $2i$ et $i$ vaut $i$. Au moins l'un des $i \leq T-1$ est divisible par la période $\tau$, ce qui assure que $x_{2i} = x_i$.

**Exemple.** $N = 11 \times 13 = 143$, donc $\sqrt{N} \lesssim 12$ et $\sqrt{p} = \sqrt{11} \lesssim 4$.

avec $x_0 = 133$

| $i$ | $x_i$ | $y_i = x_{2i}$ |
|---|---|---|
| 0 | 133 | 133 |
| 1 | 101 | 49 |
| 2 | 49 | 127 |
| 3 | 114 | 127 |
| 4 | 127 | 127 |
| 5 | 114 | 127 |
| 6 | 127 | 127 |
| 7 | 114 | 127 |
| 8 | 127 | 127 |

**période =** 2
**prépériode =** 4

avec $x_0 = 34$

| $i$ | $x_i$ | $y_i = x_{2i}$ |
|---|---|---|
| 0 | 34 | 34 |
| 1 | 13 | 27 |
| 2 | 27 | 83 |
| 3 | 15 | 105 |
| 4 | 83 | 83 |
| 5 | 26 | 105 |
| 6 | 105 | 83 |
| 7 | 15 | 105 |
| 8 | 83 | 83 |

**période =** 4
**prépériode =** 4

# Algorithme $\rho$ de Pollard

ALGORITHME $\rho$ DE POLLARD POUR LA FACTORISATION (1975)

**Entrée :** un entier $N$ composé
**Sortie :** un facteur $d$ de $N$

**Donnée externe :** une application

$$f : \mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/\mathbb{Z}$$

typiquement $f(z) = z^2 + 1 \mod N$

1. Tirer $a$ uniformément dans $\{0, \ldots, N-1\}$
2. Initialiser $x \leftarrow f(a)$ et $y \leftarrow f(f(a))$
3. Calculer $d = \text{pgcd}(y - x, N)$
4. **Tant que** $d = 1$ :
   - Calculer $x \leftarrow f(x)$
   - Calculer $y \leftarrow f(f(y))$
   - Calculer $d = \text{pgcd}(y - x, N)$
5. Si $d = N$, revenir à l'étape 1.
6. Sinon, retourner $d$.

# Algorithme $\rho$ de Pollard

ALGORITHME $\rho$ DE POLLARD POUR LA FACTORISATION (1975)

**Entrée :** un entier $N$ composé
**Sortie :** un facteur $d$ de $N$

**Donnée externe :** une application

$$f : \mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/\mathbb{Z}$$

typiquement $f(z) = z^2 + 1 \mod N$

1. Tirer $a$ uniformément dans $\{0, \ldots, N-1\}$
2. Initialiser $x \leftarrow f(a)$ et $y \leftarrow f(f(a))$
3. Calculer $d = \text{pgcd}(y - x, N)$
4. **Tant que** $d = 1$ :
   - Calculer $x \leftarrow f(x)$
   - Calculer $y \leftarrow f(f(y))$
   - Calculer $d = \text{pgcd}(y - x, N)$
5. Si $d = N$, revenir à l'étape 1.
6. Sinon, retourner $d$.

**Complexité.**

- Le nombre d'itérations de la boucle 4. est $O(\sqrt{p})$ avec grande probabilité, où $p$ est le plus petit facteur premier de $N$.
- On a $\text{pgcd}(y - x, N) = N$ avec faible probabilité : cela correspond à avoir une collision dans $\mathbb{Z}/N\mathbb{Z}$ avant d'en avoir dans des $\mathbb{Z}/p_i\mathbb{Z}$.

# Algorithme $\rho$ de Pollard

ALGORITHME $\rho$ DE POLLARD POUR LA FACTORISATION (1975)

**Entrée :** un entier $N$ composé
**Sortie :** un facteur $d$ de $N$

**Donnée externe :** une application

$$f : \mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/\mathbb{Z}$$

typiquement $f(z) = z^2 + 1 \mod N$

1. Tirer $a$ uniformément dans $\{0, \ldots, N-1\}$
2. Initialiser $x \leftarrow f(a)$ et $y \leftarrow f(f(a))$
3. Calculer $d = \text{pgcd}(y - x, N)$
4. **Tant que** $d = 1$ :
   - Calculer $x \leftarrow f(x)$
   - Calculer $y \leftarrow f(f(y))$
   - Calculer $d = \text{pgcd}(y - x, N)$
5. Si $d = N$, revenir à l'étape 1.
6. Sinon, retourner $d$.

**Complexité.**

- Le nombre d'itérations de la boucle 4. est $O(\sqrt{p})$ avec grande probabilité, où $p$ est le plus petit facteur premier de $N$.
- On a $\text{pgcd}(y - x, N) = N$ avec faible probabilité : cela correspond à avoir une collision dans $\mathbb{Z}/N\mathbb{Z}$ avant d'en avoir dans des $\mathbb{Z}/p_i\mathbb{Z}$.

$\Longrightarrow$ la **complexité totale** est en $O(\sqrt{p}\log^2 N)$ opérations dans $\mathbb{Z}/N\mathbb{Z}$.

# Pollard $\rho$ : exemple

**Exemple.** Pollard $\rho$ avec $N = 4307$

avec $x_0 = 2747$

| $x_i$ | $y_i = x_{2i}$ | $\text{pgcd}(x_i - y_i, N)$ |
|---|---|---|
| 146 | 4089 | 1 |
| 4089 | 370 | 1 |
| 148 | 3451 | 1 |
| 370 | 2027 | 1 |
| 3384 | 370 | 1 |
| 3451 | 3451 | 4307 |

Pas de chance...

avec $x_0 = 2748$

| $x_i$ | $y_i = x_{2i}$ | $\text{pgcd}(x_i - y_i, N)$ |
|---|---|---|
| 1334 | 766 | 1 |
| 766 | 2188 | 1 |
| 1005 | 1267 | 1 |
| 2188 | 620 | 1 |
| 2268 | 3502 | 1 |
| 1267 | 2435 | 73 |

C'est bon : $N = 73 \times 59$

Pour $N = 4307$, il y a 516 « mauvais » $x_0$ : proportion $\simeq 0.12$.

**Fait.** La proportion de « mauvais » $x_0$ diminue heuristiquement :

- Pour $N = 253$, il y a 98 « mauvais » $x_0$ : proportion $\simeq 0.38$.
- Pour $N = 1511057$, il y a 4378 « mauvais » $x_0$ : proportion $\simeq 0.0029$...

# Plan

# Nombres friables

**Définition.** Soit $B \geq 2$. Un entier $N \geq 2$ est $B$-friable ($B$-*smooth*) si tous ses diviseurs premiers sont plus petits que $B$.

# Nombres friables

**Définition.** Soit $B \geq 2$. Un entier $N \geq 2$ est $B$-friable ($B$-*smooth*) si tous ses diviseurs premiers sont plus petits que $B$.

**Définition.** Soit $B \geq 2$. Un entier $N \geq 2$ est $B$-superfriable ($B$-*powersmooth*) si tous les $p^e$ (avec $p$ premier) qui divisent $N$ satisfont $p^e \leq B$.

# Nombres friables

**Définition.** Soit $B \geq 2$. Un entier $N \geq 2$ est $B$-friable (*B-smooth*) si tous ses diviseurs premiers sont plus petits que $B$.

**Définition.** Soit $B \geq 2$. Un entier $N \geq 2$ est $B$-superfriable (*B-powersmooth*) si tous les $p^e$ (avec $p$ premier) qui divisent $N$ satisfont $p^e \leq B$.

**Exemple.** $N = 720 = 2^4 \times 3^2 \times 5$

- 720 est 5-friable,
- 720 n'est pas 5-superfriable, car $3^2 > 5$,
- 720 est 16-superfriable.

# Nombres friables

**Définition.** Soit $B \geq 2$. Un entier $N \geq 2$ est $B$-friable (*B-smooth*) si tous ses diviseurs premiers sont plus petits que $B$.

**Définition.** Soit $B \geq 2$. Un entier $N \geq 2$ est $B$-superfriable (*B-powersmooth*) si tous les $p^e$ (avec $p$ premier) qui divisent $N$ satisfont $p^e \leq B$.

**Exemple.** $N = 720 = 2^4 \times 3^2 \times 5$

- 720 est 5-friable,
- 720 n'est pas 5-superfriable, car $3^2 > 5$,
- 720 est 16-superfriable.

**Théorème.** Soit $C_{N,B}$ le nombre d'entiers $B$-friables entre 1 et $N$. On a asymptotiquement

$$\frac{C_{N,B}}{N} \sim u^{-u+o(1)} \quad \text{où } u = \frac{\log_2 N}{\log_2 B}$$

# Nombres friables

**Définition.** Soit $B \geq 2$. Un entier $N \geq 2$ est $B$-friable ($B$-*smooth*) si tous ses diviseurs premiers sont plus petits que $B$.

**Définition.** Soit $B \geq 2$. Un entier $N \geq 2$ est $B$-superfriable ($B$-*powersmooth*) si tous les $p^e$ (avec $p$ premier) qui divisent $N$ satisfont $p^e \leq B$.

**Exemple.** $N = 720 = 2^4 \times 3^2 \times 5$

- 720 est 5-friable,
- 720 n'est pas 5-superfriable, car $3^2 > 5$,
- 720 est 16-superfriable.

**Théorème.** Soit $C_{N,B}$ le nombre d'entiers $B$-friables entre 1 et $N$. On a asymptotiquement

$$\frac{C_{N,B}}{N} \sim u^{-u+o(1)} \quad \text{où } u = \frac{\log_2 N}{\log_2 B}$$

La probabilité que $N$ soit $N^\epsilon$-friable est donc $\varepsilon^{-\varepsilon}$.

- si $B = O(\sqrt{N})$, alors $N$ est $B$-friable avec probabilité $1/4$.
- si $B = N^{0.1}$, alors $N$ est $B$-friable avec probabilité $10^{-10}$...

**Théorème (petit théorème de Fermat).** Soit $p$ premier. Pour tout $a$ non-multiple de $p$, on a

$$a^{p-1} \equiv 1 \mod p\,.$$

**Théorème (petit théorème de Fermat).** Soit $p$ premier. Pour tout $a$ non-multiple de $p$, on a

$$a^{p-1} \equiv 1 \mod p\,.$$

**Conséquence.** Si $M$ est un multiple de $p-1$, on a également $p \mid a^M - 1$.

**Théorème (petit théorème de Fermat).** Soit $p$ premier. Pour tout $a$ non-multiple de $p$, on a

$$a^{p-1} \equiv 1 \mod p\,.$$

**Conséquence.** Si $M$ est un multiple de $p-1$, on a également $p \mid a^M - 1$.

**Idée :** on choisit $M$ un nombre $B$-superfriable, avec $B$ assez petit. Typiquement :

$$M = \text{ppcm}\{2, \ldots, B\} = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$$

où les $p_i$ sont premiers et $e_i$ maximal tel que $p_i^{e_i} \leq B$.

**Théorème (petit théorème de Fermat).** Soit $p$ premier. Pour tout $a$ non-multiple de $p$, on a

$$a^{p-1} \equiv 1 \mod p\,.$$

**Conséquence.** Si $M$ est un multiple de $p-1$, on a également $p \mid a^M - 1$.

**Idée :** on choisit $M$ un nombre $B$-superfriable, avec $B$ assez petit. Typiquement :

$$M = \text{ppcm}\{2,\ldots,B\} = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$$

où les $p_i$ sont premiers et $e_i$ maximal tel que $p_i^{e_i} \leq B$.
Puis, on espère obtenir un facteur propre de $N$ en calculant $\text{pgcd}(a^M - 1, N)$ où $a$ est tiré aléatoirement dans $\{2,\ldots,N-1\}$.

# Méthode $p-1$ de Pollard

MÉTHODE $p-1$ DE POLLARD POUR LA FACTORISATION

**Entrée :** un entier $N$ à factoriser, un entier $B \geq 2$
**Sortie :** un facteur propre de $N$
**Hypothèse :** il existe un facteur premier $p$ de $N$ tel que $p-1$ est $B$-superfriable

1. Calculer $M \leftarrow \text{ppcm}\{2, \ldots, B\}$.
2. Tirer aléatoirement $a \in \{2, \ldots, N-1\}$.
3. Calculer $d = \text{pgcd}(a, N)$.
4. **Si** $d \neq 1$, alors **retourner** $d$.
5. **Sinon :**
   - Calculer $c \leftarrow a^M - 1 \mod N$.
   - Calculer $d' \leftarrow \text{pgcd}(c, N)$.
   - **Si** $d' = N$, revenir à l'étape 2.
   - **Sinon**, **retourner** $d'$.

# Méthode $p-1$ de Pollard

MÉTHODE $p-1$ DE POLLARD POUR LA FACTORISATION

**Entrée :** un entier $N$ à factoriser, un entier $B \geq 2$
**Sortie :** un facteur propre de $N$
**Hypothèse :** il existe un facteur premier $p$ de $N$ tel que $p-1$ est $B$-superfriable

1. Calculer $M \leftarrow \text{ppcm}\{2, \ldots, B\}$.
2. Tirer aléatoirement $a \in \{2, \ldots, N-1\}$.
3. Calculer $d = \text{pgcd}(a, N)$.
4. **Si** $d \neq 1$, alors **retourner** $d$.
5. **Sinon :**
   - Calculer $c \leftarrow a^M - 1 \mod N$.
   - Calculer $d' \leftarrow \text{pgcd}(c, N)$.
   - **Si** $d' = N$, revenir à l'étape 2.
   - **Sinon**, **retourner** $d'$.

**Remarque.** Au lieu de calculer directement $\text{pgcd}(a^M - 1, N)$, on peut également calculer des pgcd « intermédiaires », de la forme $\text{pgcd}(a^{M_i} - 1, N)$ où $M_i$ est un facteur de $M$ qu'on calcule lors de l'étape 1.
$\Longrightarrow$ intéressant surtout pour de petites valeurs de $N$.

# Pollard $p-1$ : complexité

1. Calculer $M \leftarrow \text{ppcm}\{2, \ldots, B\}$.
2. Tirer aléatoirement $a \in \{1, \ldots, N-1\}$.
3. Calculer $d = \text{pgcd}(a, N)$.
4. **Si** $d \neq 1$, alors **retourner** $d$.
5. **Sinon :**
   - Calculer $c \leftarrow a^M - 1 \mod N$.
   - Calculer $d' \leftarrow \text{pgcd}(c, N)$.
   - **Si** $d' = N$, revenir à l'étape 3.
   - **Sinon**, **retourner** $d'$.

# Pollard $p-1$ : complexité

1. Calculer $M \leftarrow \text{ppcm}\{2, \ldots, B\}$.
2. Tirer aléatoirement $a \in \{1, \ldots, N-1\}$.
3. Calculer $d = \text{pgcd}(a, N)$.
4. **Si** $d \neq 1$, alors **retourner** $d$.
5. **Sinon :**
   - Calculer $c \leftarrow a^M - 1 \mod N$.
   - Calculer $d' \leftarrow \text{pgcd}(c, N)$.
   - **Si** $d' = N$, revenir à l'étape 3.
   - **Sinon**, **retourner** $d'$.

**Complexité.** L'étape coûteuse est le calcul de $a^M - 1 \mod N$, qui se fait en $O(B)$ opérations, comme $M = O(2^B)$.
$\Longrightarrow$ complexité totale $O(B \log B \log^2 N)$

# Pollard $p-1$ : complexité

1. Calculer $M \leftarrow \text{ppcm}\{2, \ldots, B\}$.
2. Tirer aléatoirement $a \in \{1, \ldots, N-1\}$.
3. Calculer $d = \text{pgcd}(a, N)$.
4. **Si** $d \neq 1$, alors **retourner** $d$.
5. **Sinon :**
   - Calculer $c \leftarrow a^M - 1 \mod N$.
   - Calculer $d' \leftarrow \text{pgcd}(c, N)$.
   - **Si** $d' = N$, revenir à l'étape 3.
   - **Sinon**, **retourner** $d'$.

**Complexité.** L'étape coûteuse est le calcul de $a^M - 1 \mod N$, qui se fait en $O(B)$ opérations, comme $M = O(2^B)$.
$\implies$ complexité totale $O(B \log B \log^2 N)$

**Remarques.** Il se peut que $d' = N$ (fréquent lorsque $N$ est petit).

- Cela peut se produire si $a$ a un petit ordre modulo $N$ (on tire $a$ à nouveau).
- Si cela se produit systématiquement, cela signifie souvent que **tous** les facteurs premiers $p$ de $N$ tels que $p-1$ est $B$-superfriable.
- Il existe d'autres cas d'échec pathologiques, rares dès lors que $N$ grandit.

# Pollard $p-1$ : exemple

Pour $N = 108\,147\,037$

# Pollard $p-1$ : exemple

Pour $N = 108\,147\,037$

- $N$ est assez petit : on exécute donc la variante avec les pgcd intermédiaires.

# Pollard $p-1$ : exemple

Pour $N = 108\,147\,037$

- $N$ est assez petit : on exécute donc la variante avec les pgcd intermédiaires.
- Pour l'exemple, on choisit $B = 14$ (on pourrait aller plus loin).

# Pollard $p-1$ : exemple

Pour $N = 108\,147\,037$

- $N$ est assez petit : on exécute donc la variante avec les pgcd intermédiaires.
- Pour l'exemple, on choisit $B = 14$ (on pourrait aller plus loin).
- Donc $M = Q_1 \times \cdots \times Q_\ell = 2^3 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 13$.

# Pollard $p-1$ : exemple

Pour $N = 108\,147\,037$

- $N$ est assez petit : on exécute donc la variante avec les pgcd intermédiaires.
- Pour l'exemple, on choisit $B = 14$ (on pourrait aller plus loin).
- Donc $M = Q_1 \times \cdots \times Q_\ell = 2^3 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 13$.
- On note $M_i = Q_1 \times \cdots \times Q_i$ (calcul intermédiaire).

# Pollard $p-1$ : exemple

Pour $N = 108\,147\,037$

- $N$ est assez petit : on exécute donc la variante avec les pgcd intermédiaires.
- Pour l'exemple, on choisit $B = 14$ (on pourrait aller plus loin).
- Donc $M = Q_1 \times \cdots \times Q_\ell = 2^3 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 13$.
- On note $M_i = Q_1 \times \cdots \times Q_i$ (calcul intermédiaire).

**Exemple** avec $a = 2$

| $Q_i$ | $a^{M_i} - 1 \mod N$ | $\text{pgcd}(a^{M_i} - 1, N)$ |
|---|---|---|
| 8 | 255 | 1 |
| 9 | 77888826 | 1 |
| 5 | 14060764 | 1 |
| 7 | 66102662 | 1 |
| 11 | 53395803 | 1 |
| 13 | 92398868 | 36037 |

On a le facteur $p = 36037$, et
$p - 1 = 2^2 \cdot 3^2 \cdot 7 \cdot 11 \cdot 13$.

# Pollard $p-1$ : exemple

Pour $N = 108\,147\,037$

- $N$ est assez petit : on exécute donc la variante avec les pgcd intermédiaires.
- Pour l'exemple, on choisit $B = 14$ (on pourrait aller plus loin).
- Donc $M = Q_1 \times \cdots \times Q_\ell = 2^3 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 13$.
- On note $M_i = Q_1 \times \cdots \times Q_i$ (calcul intermédiaire).

**Exemple** avec $a = 2$

| $Q_i$ | $a^{M_i} - 1 \mod N$ | $\mathrm{pgcd}(a^{M_i} - 1, N)$ |
|---|---|---|
| 8 | 255 | 1 |
| 9 | 77888826 | 1 |
| 5 | 14060764 | 1 |
| 7 | 66102662 | 1 |
| 11 | 53395803 | 1 |
| 13 | 92398868 | 36037 |

On a le facteur $p = 36037$, et
$p - 1 = 2^2 \cdot 3^2 \cdot 7 \cdot 11 \cdot 13$.

**Exemple** avec $a = 20$ (choisi pour l'exemple)

| $Q_i$ | $a^{M_i} - 1 \mod N$ | $\mathrm{pgcd}(a^{M_i} - 1, N)$ |
|---|---|---|
| 8 | 77299267 | 1 |
| 9 | 45988448 | 1 |
| 5 | 97367445 | 3001 |
| 7 | $\times$ | $\times$ |
| 11 | $\times$ | $\times$ |
| 13 | $\times$ | $\times$ |

On a le facteur $p = 3001$... chanceusement

# Vers une extension de la méthode $p-1$

**Intuition.** Pour la méthode $p-1$ on a utilisé implicitement le morphisme surjectif

$$\begin{array}{ccc} (\mathbb{Z}/N\mathbb{Z})^\times & \longrightarrow & (\mathbb{Z}/p\mathbb{Z})^\times \\ a & \longmapsto & a \mod p \end{array}$$

Puis, on a cherché à obtenir l'élément neutre de $(\mathbb{Z}/p\mathbb{Z})^\times$ :

$$p-1 \text{ divise } M \quad \Longleftrightarrow \quad (a \mod p)^M = 1,$$

ce que l'on peut tester en calculant $\text{pgcd}(a^M - 1, N)$.

# Vers une extension de la méthode $p-1$

**Intuition.** Pour la méthode $p-1$ on a utilisé implicitement le morphisme surjectif

$$\begin{array}{ccc} (\mathbb{Z}/N\mathbb{Z})^\times & \longrightarrow & (\mathbb{Z}/p\mathbb{Z})^\times \\ a & \longmapsto & a \mod p \end{array}$$

Puis, on a cherché à obtenir l'élément neutre de $(\mathbb{Z}/p\mathbb{Z})^\times$ :

$$p-1 \text{ divise } M \quad \Longleftrightarrow \quad (a \mod p)^M = 1,$$

ce que l'on peut tester en calculant $\mathrm{pgcd}(a^M - 1, N)$.

**Idée.** Cherchons un groupe d'un ordre différent. Pour cela, on peut même partir de plusieurs éléments de $\mathbb{Z}/N\mathbb{Z}$.

# Vers une extension de la méthode $p-1$

**Intuition.** Pour la méthode $p-1$ on a utilisé implicitement le morphisme surjectif

$$\begin{array}{ccc} (\mathbb{Z}/N\mathbb{Z})^\times & \longrightarrow & (\mathbb{Z}/p\mathbb{Z})^\times \\ a & \longmapsto & a \mod p \end{array}$$

Puis, on a cherché à obtenir l'élément neutre de $(\mathbb{Z}/p\mathbb{Z})^\times$ :

$$p-1 \text{ divise } M \quad \Longleftrightarrow \quad (a \mod p)^M = 1,$$

ce que l'on peut tester en calculant $\text{pgcd}(a^M - 1, N)$.

**Idée.** Cherchons un groupe d'un ordre différent. Pour cela, on peut même partir de plusieurs éléments de $\mathbb{Z}/N\mathbb{Z}$.

Par exemple, on se fixe $D \in \mathbb{Z}$ et on considère

$$\mathcal{A} = \{(x,y) \in (\mathbb{Z}/N\mathbb{Z})^2 \mid x^2 - Dy^2 = 1\} \subseteq (\mathbb{Z}/N\mathbb{Z})^2$$

Alors on peut considérer $\mathcal{A}_p := \mathcal{A} \mod p$ :

$$\begin{array}{ccc} \mathcal{A} & \longrightarrow & \mathcal{A}_p \\ (x,y) & \longmapsto & (x \mod p,\ y \mod p) \end{array}$$

# Vers une extension de la méthode $p-1$

**Intuition.** Pour la méthode $p-1$ on a utilisé implicitement le morphisme surjectif

$$\begin{array}{ccc} (\mathbb{Z}/N\mathbb{Z})^\times & \longrightarrow & (\mathbb{Z}/p\mathbb{Z})^\times \\ a & \longmapsto & a \mod p \end{array}$$

Puis, on a cherché à obtenir l'élément neutre de $(\mathbb{Z}/p\mathbb{Z})^\times$ :

$$p-1 \text{ divise } M \quad \Longleftrightarrow \quad (a \mod p)^M = 1,$$

ce que l'on peut tester en calculant $\mathrm{pgcd}(a^M - 1, N)$.

**Idée.** Cherchons un groupe d'un ordre différent. Pour cela, on peut même partir de plusieurs éléments de $\mathbb{Z}/N\mathbb{Z}$.

Par exemple, on se fixe $D \in \mathbb{Z}$ et on considère

$$\mathcal{A} = \{(x,y) \in (\mathbb{Z}/N\mathbb{Z})^2 \mid x^2 - Dy^2 = 1\} \subseteq (\mathbb{Z}/N\mathbb{Z})^2$$

Alors on peut considérer $\mathcal{A}_p := \mathcal{A} \mod p$ :

$$\begin{array}{ccc} \mathcal{A} & \longrightarrow & \mathcal{A}_p \\ (x,y) & \longmapsto & (x \mod p,\ y \mod p) \end{array}$$

**Question.** Que dire de la structure algébrique de $\mathcal{A}_p$ ?

# Méthode $p+1$ de Williams

$$\begin{array}{ccc} \mathcal{A} & \longrightarrow & \mathcal{A}_p \\ (x,y) & \longmapsto & (x \mod p,\ y \mod p) \end{array}$$

**Propriété.** Si $D \in \mathbb{Z}$ n'est pas un carré modulo $p$, alors $\mathbb{F}_{p^2}$ s'écrit $\mathbb{F}_p(\sqrt{D})$, et la projection $\mathcal{A}_p := (\mathcal{A} \bmod p)$ est isomorphe à $\mathcal{U}_p = \{u \in \mathbb{F}_{p^2} \mid u^{p+1} = 1\}$ qui est un groupe d'ordre $p+1$.

$$\begin{array}{ccc} \mathcal{A} & \longrightarrow & \mathcal{A}_p \\ (x,y) & \longmapsto & (x \mod p,\ y \mod p) \end{array}$$

**Propriété.** Si $D \in \mathbb{Z}$ n'est pas un carré modulo $p$, alors $\mathbb{F}_{p^2}$ s'écrit $\mathbb{F}_p(\sqrt{D})$, et la projection $\mathcal{A}_p := (\mathcal{A} \bmod p)$ est isomorphe à $\mathcal{U}_p = \{u \in \mathbb{F}_{p^2} \mid u^{p+1} = 1\}$ qui est un groupe d'ordre $p+1$.

**Éléments de preuve.** Si $(x,y) \in \mathcal{A}_p$, alors

$$u = x + y\sqrt{D} \in \mathbb{F}_{p^2} \longleftrightarrow (x,y) \in \mathcal{A}_p$$

et

$$u^{p+1} = u^p u =$$

$$\begin{array}{ccc} \mathcal{A} & \longrightarrow & \mathcal{A}_p \\ (x,y) & \longmapsto & (x \mod p,\ y \mod p) \end{array}$$

**Propriété.** Si $D \in \mathbb{Z}$ n'est pas un carré modulo $p$, alors $\mathbb{F}_{p^2}$ s'écrit $\mathbb{F}_p(\sqrt{D})$, et la projection $\mathcal{A}_p := (\mathcal{A} \bmod p)$ est isomorphe à $\mathcal{U}_p = \{u \in \mathbb{F}_{p^2} \mid u^{p+1} = 1\}$ qui est un groupe d'ordre $p+1$.

**Éléments de preuve.** Si $(x,y) \in \mathcal{A}_p$, alors

$$u = x + y\sqrt{D} \in \mathbb{F}_{p^2} \longleftrightarrow (x,y) \in \mathcal{A}_p$$

et

$$u^{p+1} = u^p u = (x + y\sqrt{D}^p)(x + y\sqrt{D}) =$$

$$\begin{array}{ccc} \mathcal{A} & \longrightarrow & \mathcal{A}_p \\ (x,y) & \longmapsto & (x \mod p,\ y \mod p) \end{array}$$

**Propriété.** Si $D \in \mathbb{Z}$ n'est pas un carré modulo $p$, alors $\mathbb{F}_{p^2}$ s'écrit $\mathbb{F}_p(\sqrt{D})$, et la projection $\mathcal{A}_p := (\mathcal{A} \bmod p)$ est isomorphe à $\mathcal{U}_p = \{u \in \mathbb{F}_{p^2} \mid u^{p+1} = 1\}$ qui est un groupe d'ordre $p+1$.

**Éléments de preuve.** Si $(x,y) \in \mathcal{A}_p$, alors

$$u = x + y\sqrt{D} \in \mathbb{F}_{p^2} \longleftrightarrow (x,y) \in \mathcal{A}_p$$

et

$$u^{p+1} = u^p u = (x + y\sqrt{D}^p)(x + y\sqrt{D}) = (x + y(-\sqrt{D}))(x + y\sqrt{D}) =$$

$$\begin{array}{ccc} \mathcal{A} & \longrightarrow & \mathcal{A}_p \\ (x,y) & \longmapsto & (x \mod p,\ y \mod p) \end{array}$$

**Propriété.** Si $D \in \mathbb{Z}$ n'est pas un carré modulo $p$, alors $\mathbb{F}_{p^2}$ s'écrit $\mathbb{F}_p(\sqrt{D})$, et la projection $\mathcal{A}_p := (\mathcal{A} \bmod p)$ est isomorphe à $\mathcal{U}_p = \{u \in \mathbb{F}_{p^2} \mid u^{p+1} = 1\}$ qui est un groupe d'ordre $p+1$.

**Éléments de preuve.** Si $(x,y) \in \mathcal{A}_p$, alors

$$u = x + y\sqrt{D} \in \mathbb{F}_{p^2} \longleftrightarrow (x,y) \in \mathcal{A}_p$$

et

$$u^{p+1} = u^p u = (x + y\sqrt{D}^p)(x + y\sqrt{D}) = (x + y(-\sqrt{D}))(x + y\sqrt{D}) = x^2 - y^2 D = 1\,.$$

$$\begin{array}{ccc} \mathcal{A} & \longrightarrow & \mathcal{A}_p \\ (x,y) & \longmapsto & (x \mod p,\ y \mod p) \end{array}$$

**Propriété.** Si $D \in \mathbb{Z}$ n'est pas un carré modulo $p$, alors $\mathbb{F}_{p^2}$ s'écrit $\mathbb{F}_p(\sqrt{D})$, et la projection $\mathcal{A}_p \coloneqq (\mathcal{A} \bmod p)$ est isomorphe à $\mathcal{U}_p = \{u \in \mathbb{F}_{p^2} \mid u^{p+1} = 1\}$ qui est un groupe d'ordre $p+1$.

**Éléments de preuve.** Si $(x,y) \in \mathcal{A}_p$, alors

$$u = x + y\sqrt{D} \in \mathbb{F}_{p^2} \longleftrightarrow (x,y) \in \mathcal{A}_p$$

et

$$u^{p+1} = u^p u = (x + y\sqrt{D}^p)(x + y\sqrt{D}) = (x + y(-\sqrt{D}))(x + y\sqrt{D}) = x^2 - y^2 D = 1\,.$$

Par ailleurs $u = 1$ est l'unique élément de $\mathcal{U}_p$ tel que $x = 1$.

# Méthode $p+1$ de Williams

$$\begin{array}{ccc} \mathcal{A} & \longrightarrow & \mathcal{A}_p \\ (x,y) & \longmapsto & (x \mod p,\ y \mod p) \end{array}$$

**Propriété.** Si $D \in \mathbb{Z}$ n'est pas un carré modulo $p$, alors $\mathbb{F}_{p^2}$ s'écrit $\mathbb{F}_p(\sqrt{D})$, et la projection $\mathcal{A}_p \coloneqq (\mathcal{A} \bmod p)$ est isomorphe à $\mathcal{U}_p = \{u \in \mathbb{F}_{p^2} \mid u^{p+1} = 1\}$ qui est un groupe d'ordre $p+1$.

**Éléments de preuve.** Si $(x,y) \in \mathcal{A}_p$, alors

$$u = x + y\sqrt{D} \in \mathbb{F}_{p^2} \longleftrightarrow (x,y) \in \mathcal{A}_p$$

et

$$u^{p+1} = u^p u = (x + y\sqrt{D}^p)(x + y\sqrt{D}) = (x + y(-\sqrt{D}))(x + y\sqrt{D}) = x^2 - y^2 D = 1\,.$$

Par ailleurs $u = 1$ est l'unique élément de $\mathcal{U}_p$ tel que $x = 1$.

**Par analogie avec la méthode** $p-1$, on va donc calculer $u^M = x_M + y_M\sqrt{D}$ et espérer que $\operatorname{pgcd}(x_M - 1, N)$ donne un facteur propre de $N$.

# Méthode $p+1$ de Williams

$$\begin{array}{ccc} \mathcal{A} & \longrightarrow & \mathcal{A}_p \\ (x,y) & \longmapsto & (x \mod p,\ y \mod p) \end{array}$$

**Propriété.** Si $D \in \mathbb{Z}$ n'est pas un carré modulo $p$, alors $\mathbb{F}_{p^2}$ s'écrit $\mathbb{F}_p(\sqrt{D})$, et la projection $\mathcal{A}_p \coloneqq (\mathcal{A} \bmod p)$ est isomorphe à $\mathcal{U}_p = \{u \in \mathbb{F}_{p^2} \mid u^{p+1} = 1\}$ qui est un groupe d'ordre $p+1$.

**Éléments de preuve.** Si $(x,y) \in \mathcal{A}_p$, alors

$$u = x + y\sqrt{D} \in \mathbb{F}_{p^2} \longleftrightarrow (x,y) \in \mathcal{A}_p$$

et

$$u^{p+1} = u^p u = (x + y\sqrt{D}^p)(x + y\sqrt{D}) = (x + y(-\sqrt{D}))(x + y\sqrt{D}) = x^2 - y^2 D = 1\,.$$

Par ailleurs $u = 1$ est l'unique élément de $\mathcal{U}_p$ tel que $x = 1$.

**Par analogie avec la méthode** $p-1$, on va donc calculer $u^M = x_M + y_M\sqrt{D}$ et espérer que $\mathrm{pgcd}(x_M - 1, N)$ donne un facteur propre de $N$.

**Problème :** comment calculer $x_M$ sans connaissance de $p$ ?

# Méthode $p+1$ de Williams

**Problème :** comment calculer $x_M$ sans connaissance de $p$ ?

# Méthode $p+1$ de Williams

**Problème :** comment calculer $x_M$ sans connaissance de $p$ ?

On cherche exprimer $x_{2n}$ et $x_{2n+1}$ en fonction de précédents $x_i$ (*idée : exponentiation binaire*).

Si $(x_n, y_n)$ est défini par $u^n = x_n + y_n\sqrt{D}$, alors :

# Méthode $p+1$ de Williams

**Problème :** comment calculer $x_M$ sans connaissance de $p$ ?

On cherche exprimer $x_{2n}$ et $x_{2n+1}$ en fonction de précédents $x_i$ (*idée : exponentiation binaire*).

Si $(x_n, y_n)$ est défini par $u^n = x_n + y_n\sqrt{D}$, alors :

- On a $u^{2n} = (x_n + y_n\sqrt{D})^2 =$

# Méthode $p+1$ de Williams

**Problème :** comment calculer $x_M$ sans connaissance de $p$ ?

On cherche exprimer $x_{2n}$ et $x_{2n+1}$ en fonction de précédents $x_i$ (*idée : exponentiation binaire*).

Si $(x_n, y_n)$ est défini par $u^n = x_n + y_n\sqrt{D}$, alors :

- On a $u^{2n} = (x_n + y_n\sqrt{D})^2 = (x_n^2 + Dy_n^2) + 2x_n y_n\sqrt{D} =$

# Méthode $p+1$ de Williams

**Problème :** comment calculer $x_M$ sans connaissance de $p$ ?

On cherche exprimer $x_{2n}$ et $x_{2n+1}$ en fonction de précédents $x_i$ (*idée : exponentiation binaire*).

Si $(x_n, y_n)$ est défini par $u^n = x_n + y_n\sqrt{D}$, alors :

- On a $u^{2n} = (x_n + y_n\sqrt{D})^2 = (x_n^2 + Dy_n^2) + 2x_n y_n\sqrt{D} = (-1 + 2x_n^2) + 2x_n y_n\sqrt{D}$.

# Méthode $p+1$ de Williams

**Problème :** comment calculer $x_M$ sans connaissance de $p$ ?

On cherche exprimer $x_{2n}$ et $x_{2n+1}$ en fonction de précédents $x_i$ (*idée : exponentiation binaire*).

Si $(x_n, y_n)$ est défini par $u^n = x_n + y_n\sqrt{D}$, alors :

- On a $u^{2n} = (x_n + y_n\sqrt{D})^2 = (x_n^2 + Dy_n^2) + 2x_n y_n\sqrt{D} = (-1 + 2x_n^2) + 2x_n y_n\sqrt{D}$.
  Donc, $x_{2n} = -1 + 2x_n^2$ et $y_{2n} = 2x_n y_n$.

# Méthode $p+1$ de Williams

**Problème :** comment calculer $x_M$ sans connaissance de $p$ ?

On cherche exprimer $x_{2n}$ et $x_{2n+1}$ en fonction de précédents $x_i$ (*idée : exponentiation binaire*).

Si $(x_n, y_n)$ est défini par $u^n = x_n + y_n\sqrt{D}$, alors :

- On a $u^{2n} = (x_n + y_n\sqrt{D})^2 = (x_n^2 + Dy_n^2) + 2x_n y_n\sqrt{D} = (-1 + 2x_n^2) + 2x_n y_n\sqrt{D}$.
  Donc, $x_{2n} = -1 + 2x_n^2$ et $y_{2n} = 2x_n y_n$.
- D'une part, $u^{2n+1} = u^{2n}u$ donne $x_{2n+1} = x_{2n}x_1 + y_{2n}y_1 D = (1 + 2x_n^2)x_1 + 2x_n y_n y_1 D$.

# Méthode $p+1$ de Williams

**Problème :** comment calculer $x_M$ sans connaissance de $p$ ?

On cherche exprimer $x_{2n}$ et $x_{2n+1}$ en fonction de précédents $x_i$ (*idée : exponentiation binaire*).

Si $(x_n, y_n)$ est défini par $u^n = x_n + y_n\sqrt{D}$, alors :

- On a $u^{2n} = (x_n + y_n\sqrt{D})^2 = (x_n^2 + Dy_n^2) + 2x_n y_n\sqrt{D} = (-1 + 2x_n^2) + 2x_n y_n\sqrt{D}$.
  Donc, $x_{2n} = -1 + 2x_n^2$ et $y_{2n} = 2x_n y_n$.
- D'une part, $u^{2n+1} = u^{2n}u$ donne $x_{2n+1} = x_{2n}x_1 + y_{2n}y_1 D = (1 + 2x_n^2)x_1 + 2x_n y_n y_1 D$.
- D'autre part, $u^{n+1} = u^n u$ donne $x_{n+1} = x_n x_1 + D y_n y_1$.

# Méthode $p+1$ de Williams

**Problème :** comment calculer $x_M$ sans connaissance de $p$ ?

On cherche exprimer $x_{2n}$ et $x_{2n+1}$ en fonction de précédents $x_i$ (*idée : exponentiation binaire*).

Si $(x_n, y_n)$ est défini par $u^n = x_n + y_n\sqrt{D}$, alors :

- On a $u^{2n} = (x_n + y_n\sqrt{D})^2 = (x_n^2 + Dy_n^2) + 2x_n y_n\sqrt{D} = (-1 + 2x_n^2) + 2x_n y_n\sqrt{D}$.
  Donc, $x_{2n} = -1 + 2x_n^2$ et $y_{2n} = 2x_n y_n$.
- D'une part, $u^{2n+1} = u^{2n}u$ donne $x_{2n+1} = x_{2n}x_1 + y_{2n}y_1 D = (1 + 2x_n^2)x_1 + 2x_n y_n y_1 D$.
- D'autre part, $u^{n+1} = u^n u$ donne $x_{n+1} = x_n x_1 + D y_n y_1$.
- Donc, $x_{2n+1} = (-1 + 2x_n^2)x_1 + 2x_n(x_{n+1} - x_n x_1) = 2x_n x_{n+1} - x_1$.

# Méthode $p+1$ de Williams

**Problème :** comment calculer $x_M$ sans connaissance de $p$ ?

On cherche exprimer $x_{2n}$ et $x_{2n+1}$ en fonction de précédents $x_i$ (*idée : exponentiation binaire*).

Si $(x_n, y_n)$ est défini par $u^n = x_n + y_n\sqrt{D}$, alors :

- On a $u^{2n} = (x_n + y_n\sqrt{D})^2 = (x_n^2 + Dy_n^2) + 2x_n y_n\sqrt{D} = (-1 + 2x_n^2) + 2x_n y_n\sqrt{D}$.
  Donc, $x_{2n} = -1 + 2x_n^2$ et $y_{2n} = 2x_n y_n$.
- D'une part, $u^{2n+1} = u^{2n}u$ donne $x_{2n+1} = x_{2n}x_1 + y_{2n}y_1 D = (1 + 2x_n^2)x_1 + 2x_n y_n y_1 D$.
- D'autre part, $u^{n+1} = u^n u$ donne $x_{n+1} = x_n x_1 + D y_n y_1$.
- Donc, $x_{2n+1} = (-1 + 2x_n^2)x_1 + 2x_n(x_{n+1} - x_n x_1) = 2x_n x_{n+1} - x_1$.
- Enfin, $x_{2n+2} = -1 + 2x_{n+1}^2$.

# Méthode $p+1$ de Williams

**Problème :** comment calculer $x_M$ sans connaissance de $p$ ?

On cherche exprimer $x_{2n}$ et $x_{2n+1}$ en fonction de précédents $x_i$ (*idée : exponentiation binaire*).

Si $(x_n, y_n)$ est défini par $u^n = x_n + y_n\sqrt{D}$, alors :

- On a $u^{2n} = (x_n + y_n\sqrt{D})^2 = (x_n^2 + Dy_n^2) + 2x_n y_n\sqrt{D} = (-1 + 2x_n^2) + 2x_n y_n\sqrt{D}$.
  Donc, $x_{2n} = -1 + 2x_n^2$ et $y_{2n} = 2x_n y_n$.
- D'une part, $u^{2n+1} = u^{2n}u$ donne $x_{2n+1} = x_{2n}x_1 + y_{2n}y_1 D = (1 + 2x_n^2)x_1 + 2x_n y_n y_1 D$.
- D'autre part, $u^{n+1} = u^n u$ donne $x_{n+1} = x_n x_1 + D y_n y_1$.
- Donc, $x_{2n+1} = (-1 + 2x_n^2)x_1 + 2x_n(x_{n+1} - x_n x_1) = 2x_n x_{n+1} - x_1$.
- Enfin, $x_{2n+2} = -1 + 2x_{n+1}^2$.

Donc, la connaissance de $(x_n, x_{n+1})$ permet de déduire $(x_{2n}, x_{2n+1}, x_{2n+2})$.

# Méthode $p+1$ de Williams

**Problème :** comment calculer $x_M$ sans connaissance de $p$ ?

On cherche exprimer $x_{2n}$ et $x_{2n+1}$ en fonction de précédents $x_i$ (*idée : exponentiation binaire*).

Si $(x_n, y_n)$ est défini par $u^n = x_n + y_n\sqrt{D}$, alors :

- On a $u^{2n} = (x_n + y_n\sqrt{D})^2 = (x_n^2 + Dy_n^2) + 2x_n y_n\sqrt{D} = (-1 + 2x_n^2) + 2x_n y_n\sqrt{D}$.
  Donc, $x_{2n} = -1 + 2x_n^2$ et $y_{2n} = 2x_n y_n$.
- D'une part, $u^{2n+1} = u^{2n}u$ donne $x_{2n+1} = x_{2n}x_1 + y_{2n}y_1 D = (1 + 2x_n^2)x_1 + 2x_n y_n y_1 D$.
- D'autre part, $u^{n+1} = u^n u$ donne $x_{n+1} = x_n x_1 + D y_n y_1$.
- Donc, $x_{2n+1} = (-1 + 2x_n^2)x_1 + 2x_n(x_{n+1} - x_n x_1) = 2x_n x_{n+1} - x_1$.
- Enfin, $x_{2n+2} = -1 + 2x_{n+1}^2$.

Donc, la connaissance de $(x_n, x_{n+1})$ permet de déduire $(x_{2n}, x_{2n+1}, x_{2n+2})$.

**Conséquence :** on peut adapter l'exponentiation binaire pour calculer les $(x_n)$ !

# Méthode $p+1$ de Williams

MÉTHODE $p+1$ DE WILLIAMS POUR LA FACTORISATION (1982)
VERSION PÉDAGOGIQUE

**Entrée :** un entier $N$ à factoriser, un entier $B \geq 2$
**Sortie :** un facteur propre de $N$
**Hypothèse :** il existe un facteur premier $p$ de $N$ tel que $p+1$ est $B$-superfriable

1. Calculer $M \leftarrow \text{ppcm}\{2, \ldots, B\}$.
2. Tirer aléatoirement $x_1$ dans $\{1, \ldots, N-1\}$.
3. Calculer $d = \text{pgcd}(x_1, N)$.
4. **Si** $d \neq 1$, alors **retourner** $d$.
5. **Sinon :**
   - Calculer $x_M$ via les relations décrites précédemment.
   - Calculer $d' \leftarrow \text{pgcd}(x_M - 1, N)$.
   - **Si** $d' = N$, revenir à l'étape 2.
   - **Sinon**, **retourner** $d'$.

# Méthode $p+1$ de Williams

MÉTHODE $p+1$ DE WILLIAMS POUR LA FACTORISATION (1982)
VERSION PÉDAGOGIQUE

**Entrée :** un entier $N$ à factoriser, un entier $B \geq 2$
**Sortie :** un facteur propre de $N$
**Hypothèse :** il existe un facteur premier $p$ de $N$ tel que $p+1$ est $B$-superfriable

1. Calculer $M \leftarrow \text{ppcm}\{2, \ldots, B\}$.
2. Tirer aléatoirement $x_1$ dans $\{1, \ldots, N-1\}$.
3. Calculer $d = \text{pgcd}(x_1, N)$.
4. **Si** $d \neq 1$, alors **retourner** $d$.
5. **Sinon :**
   - Calculer $x_M$ via les relations décrites précédemment.
   - Calculer $d' \leftarrow \text{pgcd}(x_M - 1, N)$.
   - **Si** $d' = N$, revenir à l'étape 2.
   - **Sinon**, **retourner** $d'$.

**Remarque.**

- En initialisant un $x_1$ aléatoire, on détermine implicitement $D$. Avec bonne probabilité, $D$ sera un non-résidu quadratique modulo $p$.
- En pratique, on calcule $v_n = 2x_n$ au lieu de $x_n$ pour accélérer les calculs.

# Méthode $p+1$ de Williams

MÉTHODE $p+1$ DE WILLIAMS POUR LA FACTORISATION (1982)
VERSION PÉDAGOGIQUE

**Entrée :** un entier $N$ à factoriser, un entier $B \geq 2$
**Sortie :** un facteur propre de $N$
**Hypothèse :** il existe un facteur premier $p$ de $N$ tel que $p+1$ est $B$-superfriable

1. Calculer $M \leftarrow \text{ppcm}\{2, \ldots, B\}$.
2. Tirer aléatoirement $x_1$ dans $\{1, \ldots, N-1\}$.
3. Calculer $d = \text{pgcd}(x_1, N)$.
4. **Si** $d \neq 1$, alors **retourner** $d$.
5. **Sinon :**
   - Calculer $x_M$ via les relations décrites précédemment.
   - Calculer $d' \leftarrow \text{pgcd}(x_M - 1, N)$.
   - **Si** $d' = N$, revenir à l'étape 2.
   - **Sinon**, **retourner** $d'$.

**Remarque.**

- En initialisant un $x_1$ aléatoire, on détermine implicitement $D$. Avec bonne probabilité, $D$ sera un non-résidu quadratique modulo $p$.
- En pratique, on calcule $v_n = 2x_n$ au lieu de $x_n$ pour accélérer les calculs.

*A p+1 method of factoring*. Williams, H. C.. Mathematics of Computation, 39 (159). **1982**.

**Problème :** les méthodes $p-1$ et $p+1$ ne fonctionnent que si ces derniers sont friables ou superfriables.

**Problème :** les méthodes $p-1$ et $p+1$ ne fonctionnent que si ces derniers sont friables ou superfriables.

**Question.** Des variantes (ou une généralisation) pour d'autres $p+\varepsilon$ ?

**Problème :** les méthodes $p-1$ et $p+1$ ne fonctionnent que si ces derniers sont friables ou superfriables.

**Question.** Des variantes (ou une généralisation) pour d'autres $p+\varepsilon$ ?

Prenons de la **hauteur** sur les idées précédentes ($p-1$, $p+1$) :

- On construit une structure $\mathcal{E}$ sur $\mathbb{Z}/N\mathbb{Z}$, telle que **sa réduction $\mathcal{E}_p$ dans $\mathbb{F}_p$ est un groupe**.
- On sait calculer dans $\mathcal{E}_p$ via des opérations dans $\mathbb{Z}/N\mathbb{Z}$, **sans connaissance** de $p$.
- On sait (avec bonne proba) **identifier l'élément neutre** de $\mathcal{E}_p$.

# Méthode ECM de Lenstra

**Problème :** les méthodes $p-1$ et $p+1$ ne fonctionnent que si ces derniers sont friables ou superfriables.

**Question.** Des variantes (ou une généralisation) pour d'autres $p+\varepsilon$ ?

Prenons de la **hauteur** sur les idées précédentes ($p-1$, $p+1$) :

- On construit une structure $\mathcal{E}$ sur $\mathbb{Z}/N\mathbb{Z}$, telle que **sa réduction $\mathcal{E}_p$ dans $\mathbb{F}_p$ est un groupe**.
- On sait calculer dans $\mathcal{E}_p$ via des opérations dans $\mathbb{Z}/N\mathbb{Z}$, **sans connaissance** de $p$.
- On sait (avec bonne proba) **identifier l'élément neutre** de $\mathcal{E}_p$.

**ECM** (elliptic curve method) par Lenstra (1987) :

- $\mathcal{E}$ est une courbe elliptique, vue dans $\mathbb{Z}/N\mathbb{Z}$.
- L'ordre de $\mathcal{E}_p$ varie entre $(p+1-2\sqrt{p})$ et $(p+1+2\sqrt{p})$ : c'est la borne de Hasse.
- On peut choisir le point initial $P$ **et** la courbe $\mathcal{E}$ au hasard. Puis on calcule le point $M \cdot P = (x_M : y_M : z_M)$, où $M \in \mathbb{N}$ correspond à la borne de friabilité $B$.
- On travaille en coordonnées projectives : le neutre est $(x : y : z) = (0 : 1 : 0)$. On sait l'identifier : on cherche lorsque $\mathrm{pgcd}(z, N) \neq 1$.

# Méthode ECM de Lenstra

**Problème :** les méthodes $p-1$ et $p+1$ ne fonctionnent que si ces derniers sont friables ou superfriables.

**Question.** Des variantes (ou une généralisation) pour d'autres $p+\varepsilon$ ?

Prenons de la **hauteur** sur les idées précédentes $(p-1, p+1)$ :

- On construit une structure $\mathcal{E}$ sur $\mathbb{Z}/N\mathbb{Z}$, telle que **sa réduction $\mathcal{E}_p$ dans $\mathbb{F}_p$ est un groupe**.
- On sait calculer dans $\mathcal{E}_p$ via des opérations dans $\mathbb{Z}/N\mathbb{Z}$, **sans connaissance** de $p$.
- On sait (avec bonne proba) **identifier l'élément neutre** de $\mathcal{E}_p$.

**ECM** (elliptic curve method) par Lenstra (1987) :

- $\mathcal{E}$ est une courbe elliptique, vue dans $\mathbb{Z}/N\mathbb{Z}$.
- L'ordre de $\mathcal{E}_p$ varie entre $(p+1-2\sqrt{p})$ et $(p+1+2\sqrt{p})$ : c'est la borne de Hasse.
- On peut choisir le point initial $P$ **et** la courbe $\mathcal{E}$ au hasard. Puis on calcule le point $M \cdot P = (x_M : y_M : z_M)$, où $M \in \mathbb{N}$ correspond à la borne de friabilité $B$.
- On travaille en coordonnées projectives : le neutre est $(x : y : z) = (0 : 1 : 0)$. On sait l'identifier : on cherche lorsque $\mathrm{pgcd}(z, N) \neq 1$.

*Factoring integers with elliptic curves*. Lenstra Jr., H. W.. Annals of Mathematics. 126 (3). **1987**.

# Résumé

| **algorithme** | **complexité** | **commentaires** |
|---|---|---|
| divisions successives | $O(p \log^2 N)$ | élimine les très petits facteurs |
| Fermat | $O(\frac{(\sqrt{N}-p)^2}{2p} \log^2 N)$ | élimine les facteurs proches de $\sqrt{N}$ |
| Pollard $\rho$ | $O(\sqrt{p} \log^2 N)$ | élimine des facteurs petits |
| Pollard $p-1$ et $p+1$ | $O(B \log B \log^2 N)$ | élimine des facteurs proches de nombres $B$-superfriables |
| ECM | $O(2^{\sqrt{2 \log p \log \log p}})$ (sous-exp) | méthode utilisée en pratique pour éliminer les facteurs moyens (<60 chiffres) |

# Résumé

| **algorithme** | **complexité** | **commentaires** |
|---|---|---|
| divisions successives | $O(p \log^2 N)$ | élimine les très petits facteurs |
| Fermat | $O(\frac{(\sqrt{N}-p)^2}{2p} \log^2 N)$ | élimine les facteurs proches de $\sqrt{N}$ |
| Pollard $\rho$ | $O(\sqrt{p} \log^2 N)$ | élimine des facteurs petits |
| Pollard $p-1$ et $p+1$ | $O(B \log B \log^2 N)$ | élimine des facteurs proches de nombres $B$-superfriables |
| ECM | $O(2^{\sqrt{2 \log p \log \log p}})$ (sous-exp) | méthode utilisée en pratique pour éliminer les facteurs moyens (<60 chiffres) |

**Implications en cryptographie.** Dans le chiffrement RSA, où $N = pq$ , il faut éviter :

1. que $p$ ou $q$ soient petits,
2. que $p$ et $q$ soient très proches,
3. que $p + \varepsilon$ et $q + \varepsilon$ soient friables.

En pratique, la troisième condition est vérifiée avec très grande probabilité lorsque $p$ et $q$ sont de très grands premiers (plusieurs centaines de chiffres).

# Autres références


- *A Course in Computational Algebraic Number Theory*. H. Cohen. GTM38, Springer-Verlag. **1993**.
- *Prime Numbers and Computer Methods for Factorization*. H. Riesel. Progress in Mathematics, Birkhaüser. **1985**.
- *Prime Numbers, a Computational Perspective*. R. Crandall, C. Pomerance. Springer. **2001**.

# Fin du cours

**Questions ?**