Factorisation des grands nombres

Decker Benjamin - Le Boulc'h Erin

Double Licence Mathématiques Informatique - Semestre 6

Sommaire

1 Introduction

Déterminer si un nombre est premier est plus simple que de trouver les facteurs premiers de sa décomposition. On peut être convaincu moralement qu'un nombre est premier/ ou qu'il est décomposable en facteurs premiers non triviaux sans les connaître.

Objectif : Explorer les méthodes de factorisation, notamment pour de grands nombres

2 Factorisation naïve par divisions successives

Principe : trouver le plus petit diviseur de n, i.e diviser n par tous les nombres premiers plus petits que lui et voir si un d'eux le divise, puis itérer la méthode sur le quotient, jusqu'à arriver à 1. On récupère les diviseurs et on obtient la factorisation de n.

Exemple Soit n = 28.
28÷**2** = 14
14÷**2** = 7
7÷**7** = 1
On obtient donc $28 = 2 \times 2 \times 7$

Lemme Le plus petit diviseur de N ne dépasse pas $\sqrt{N}$.
En effet, prenons p le plus petit diviseur de N. Alors $N = p \times q$ avec q un entier ($p \leq q$ par minimalité de p). On a $p \leq p \times q = N$ donc $p \leq \sqrt{N}$.

Critère de finitude Soit N l'entier que l'on veut factoriser.
Si N est premier, alors, l'algorithme se finit quand on a testé tous les nombres premiers inférieurs à N : aucun diviseur n'ayant été trouvé, c'est comme cela qu'on en déduit qu'il est premier, i.e qu'il n'a pas de diviseur premier autre que 1 et lui-même.
Si N n'est pas premier, alors il est divisible par plusieurs entiers premiers plus petits que lui. La méthode permet de les trouver dans l'ordre croissant et si on considère les quotients successifs comme une suite, celle-ci est strictement décroissante et minorée par 1 donc elle est finie.

Optimisation

- On peut effectuer l'algorithme sur les $p - 1$ diviseurs plus petits de N (avec p le nombre de facteurs premiers dans la factorisation de N) et ajouter le dernier quotient (qui est un entier premier) ce qui permet de ne tester que les entiers qui ne dépassent pas la racine du quotient à chaque itération.

  Si $N = p_1 \times ... \times p_q$ avec $p_1 \leq ... \leq p_q$ premiers, pour $i \in \{1, ..., q\text{-}1\}$, p_i est le plus petit diviseur premier de $l = p_i \times .... \times p_q$ donc p_i ne dépasse pas $\sqrt{(l)}$.

- Si $d|N$, il faut ajouter d aux diviseurs et continuer l'algorithme sur le quotient de la division. Si $d \nmid N$, on sait que tous ses multiples ne divisent pas N non plus. Si on teste chaque diviseur, le nombre de division est linéaire. Ainsi, pour de très grands nombres, il pourrait être intéressant de ne pas faire de divisions inutiles. Ici, si on a une liste des entiers premiers à tester, on pourrait donc retirer ces multiples. Pour avoir une réelle optimisation, il faudrait faire attention à la suppression de ces valeurs de la liste. En effet, si on ne fait pas cette opération avec une complexité en temps constant, on ne gagne rien, voire on fait plus d'opérations que si on laissait les divisions inutiles.

  On se rend en même temps compte que pour des nombres très grands, stocker $\sqrt{N}$ entiers de plus en plus grands n'est pas viable. Il faut donc trouver une alternative.

- Si on utilise une variable qu'on incrémente à chaque fois au lieu de stocker les entiers dans une liste, on se retrouve dans le cas où on fait des divisions inutiles (potentiellement BEAUCOUP de divisions inutiles).

Quand on fait cette méthode sur de petits entiers comme dans l'exemple, c'est assez facile car on connait par coeur les petits nombres premiers ou on peut y avoir facilement accès, or la question suivante se pose pour les entiers très (très) grands : comment savoir par quels entiers diviser pour espérer trouver un diviseur premier ? Faut-il diviser par tous les entiers (pas forcément premiers) plus petits que $\sqrt{N}$, en utilisant un compteur par exemple, au risque d'avoir un très grand nombre de divisions à effectuer (il y a beaucoup plus de nombres entiers que de nombres premiers) ? Faut-il stocker au préalable dans une liste les nombres entiers inférieurs à $\sqrt{N}$ pour réduire le nombre de divisions nécessaires ? Et dans ce cas là, comment établir cette liste (Crible d'Erathostène, application des divisions successives à chaque entier intermédiaire) avec une complexité acceptable dans tous les cas ? (Dans un cas extrême, pour une puissance de 2 très grande, cette méthode serait inefficace car il suffirait de diviser successivement par 2, au lieu de construire cette liste).

Conclusion On observe que sur le principe, utiliser les divisions successives pour établir une factorisation naïve fonctionne toujours (avec le critère de finitude). Cependant, cette méthode est limitée en application pour de très grands entiers, soit en termes d'opérations, soit en termes d'espace et il n'est pas possible de trouver un compromis efficace entre les deux complexités.

3 Méthode de Fermat

Lemmes nécessaires

- Tous les nombres premiers supérieurs à 2 sont impairs.
- Une somme ou une différence de nombres impairs est paire.
- Un entier se décompose comme $2^k N$ avec $k \in \mathbb{N}$, N un entier impair. Si N est premier, on a donc la décomposition en facteurs premiers, sinon, N s'exprime comme une différence de carrés.
- Soit p, $q \in \mathbb{N}$, $pq = \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2$

Méthode On sait qu'un entier est décomposable comme $2^k N$, avec N un entier impair. Si N est premier, on a déjà la décomposition en facteurs premiers qui nous intéresse. Sinon, nous allons décomposer N en produit de deux facteurs impairs (auxquels on appliquera la même procédure s'ils ne sont pas premiers).

On veut trouver S et R tel que $N = R^2 - S^2 = (S + R)(S - R)$ Pour cela, on commence par poser $c = \lfloor \sqrt{(N)} \rfloor$ et on va y ajouter $k \in \mathbb{N}$ de plus en plus grand, jusqu'à trouver un S^2 convenable. En effet, on pose $R = c + k$ (R est forcément plus grand que $\sqrt{N}$) pour avoir $S^2 = N - R^2 = N - (c + k)^2$. Un S^2 convenable est donc tel que $S \in \mathbb{N}$. Une fois S et R trouvés, on pourra donc écrire $N = (S + R)(S - R)$

Exemple Factorisons 15 par la méthode de Fermat. (On sait que $15 = 3 \times 5$)
15 étant impair, on a $15 = 2^0 N$ avec $N = 15$
$c = \lfloor \sqrt{(15)} \rfloor = 3$
Si on prend $R = c + 0 = 3$, alors $R^2 = 9$ et $S^2 = N - R^2 = 16 - 9 = 7$. Or $\sqrt{7}$ n'est pas un entier.
On continue avec $R = c + 1 = 4$. On a $R^2 = 16$ et $S = N - R = 16 - 15 = 1$.
On a donc $S = 1$ qui est $\sqrt{(1)}$
On retrouve bien $N = (R + S)(R - S) = (4 + 1) \times (4 - 1) = 5 \times 3$