La compression des nombres premiers

par
Simon Plouffe
15 juin 2021

Résumé

Nous présentons ici un algorithme permettant de compresser les nombres premiers jusqu'à une valeur assez grande (18361375334787046697). L'algorithme utilise la représentation de la somme,

$$\sum_{n=1}^{\infty} \frac{p_n}{1000^n} = 0.002003005007011013017019023 0\ldots$$

Où p_n est le nième nombre premier.
Cette construction permet d'obtenir un réel qui est ensuite décomposé ou décrypté permettant de reconstruire la suite des premiers à partir de 2.

Abstract

We present here an algorithm that permits the compression of the primes up to a large value (18361375334787046697), the algorithm uses a representation of this sum

$$\sum_{n=1}^{\infty} \frac{p_n}{1000^n} = 0.002003005007011013017019023 0\ldots$$

p_n being the n'th prime.
Then, by using the algorithm, one can reconstruct the series of primes from 2.

Introduction

Pour illustrer l'algorithme prenon l'exemple avec b = 10, on a donc la somme :

$$\sum_{n=1}^{\infty} \frac{p_n}{10^n} = 0.2358249162515829584918287 \ldots$$

Qui donne de façon équivalente la suite : a_n

[2, 3, 5, 8, 2, 4, 9, 1, 6, 2, 5, 1, 5, 8, 2, 9, 5, 8, 4, 9, 1, 8, 2, 8, 7, …]

On utilisera ensuite un algorithme pour modifier chaque élément de la suite de manière à obtenir une suite strictement croissante à l'aide de 2 règles. Ici la base est b=10.

1) Si $a_{n+1} \geq a_n$ alors $a_n = a_n - 1$ et $a_{n+1} = b + a_{n+1}$
2) Si $a_n \neq 2$ et $a_n = 0 \bmod 2$ alors $a_n = a_n - 1$ et $a_{n+1} = b + a_{n+1}$

Donc, après le premier passage on obtient la suite :

[2, 3, 5, 7, 11, 13, 17, 19, 25, 11, 13, 19, 23, 27, 11, 17, 23, 27, 13, 19, …]

Après un 2ème passage, la suite devient :

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 45, 29, 35, 41, 45, 31, 39, …]

Finalement, elle devient :

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 79, …]

Notons que cette dernière suite a les mêmes propriétés en regard de la somme originale. Le déplacement des chiffres de la suite ne change en rien la somme au final.

En fait, les nombres premiers pris individuellement ou sous forme de réel écrit en base 10 est la même information. L'algorithme ne permet que de décoder le nombre réel construit.

Notons également que l'algorithme est valide pour la base 10 jusqu'à ce que l'écart entre 2 premiers dépasse 2b = 20, en effet à 1129 (le 189ème premier), l'écart est de 22 jusqu'à 1151 et c'est justement à cet endroit précis que l'algorithme fait défaut.

En partant de ce principe que : tant que l'écart entre 2 premiers est plus petit que 2b, alors c'est vérifié. En effet pour la base 1000, en consultant les records connus d'écarts entre

les nombres premiers par ordre croissant on constate qu'en utilisant la base 1000 on peut se rendre jusqu'à 18361375334787046697 [voir réf. 2].

Taux de compression

En prenant une table de tous les premiers jusqu'à 9999999999971, soit tous les premiers de 13 chiffres et moins, l'espace requis est exactement de 4802679133320 octets (4.8 TB). Le codage en utilisant la base 1000 demande à raison de 3 chiffres par premier un fichier de $346065536839 * 3 = 1.03$ TB, puisqu'il y a 346065536839 nombres premiers. On a donc une compression de l'ordre de 21,61 % ou un ratio de 4,61.

Le programme gzip ou gunzip permet de faire au mieux 33 %. Si en plus le fichier texte obtenu grâce au codage par un nombre réel est compressé le gain est le l'ordre d'un facteur de 10.

À toutes fin pratique, la base 1000 est optimale, d'abord par sa simplicité d'utilisation et aussi le fait qu'on puisse se rendre jusqu'à 1.83 x 10^{19}. Un calcul rapide pour une table encodée des premiers occuperait un espace assez important mais aussi un gain de 90 % (10 % de la taille originale).

Le borne de 1.83 x 10^{19} est la plus grande connue où l'écart est de 1550 entre 2 premiers successifs, puisqu'on peut se rendre à 2000 comme écart il est raisonnable de penser qu'on puisse encoder les premiers jusqu'à 10^{20}.

Temps de calcul

L'algorithme présenté (page suivante) est naïf et demande n passages pour être certain que la suite ne bouge plus, il est certainement possible de l'améliorer. Par exemple, il n'est pas nécessaire de faire un passage de 1 à n à chaque fois puisque si a_n et a_{n+1} sont impairs et que $a_{n+1} > a_n$, les 2 termes de croitront plus.

La représentation par nombre réel est utilisée ici pour faciliter les calculs mais on pourrait très bien tout faire en utlisant que des entiers de 32 bits.

On peut également coder à partir d'un point donné M, il suffit d'avoir le dernier premier connu avant M et de construire le réel (ou suite de gros entiers) à partir de ce point.

Méthode	Commentaire	Taux de compression	Constante
Concaténation	Nombre irrationnel, constante de Copeland-Erdos	1	0.2357911131719
Fridman et al. (2019)		Varie entre 0,8 et 0,95	2.920050977316
Base 10	Valide jusqu'à 1129	0,34	0.2358249162515829
Base 100	Valide jusqu'à 20831323	0,28	0.02030507111317192
Base 1000	Valide jusqu'à 18361375334787046697	Entre 0,10 et 0,21	0.002003005007011013

Algorithm (Maple)

```
listp := proc(s, b, m)
# s is the sequence of numbers from the base b expansion.
# b : base
# m : number of elements of the base.

local aa, n, ll, nn;
   aa := Array(s);
   for n to m do
      if aa[n + 1] <= aa[n] then
         aa[n] := aa[n] - 1;
         aa[n + 1] := b + aa[n + 1]
      end if;
      if aa[n] <> 2 and aa[n] mod 2 = 0 then
         aa[n] := aa[n] - 1;
         aa[n + 1] := b + aa[n + 1]
      end if
   end do;
   return aa
# return de Array : printing is necessary to examine the sequence.
end;
```

References

[1] N.J.A. Sloane, Simon Plouffe, *Encyclopedia of Integer Sequences,* Academic Press , San Diego 1995.

[2] Prime gap : https://en.wikipedia.org/wiki/Prime_gap

[3] The OEIS : http://oeis.org/, sequences : A005669, A002386, A005250

[4] Plouffe, Simon, *A set of formulas for primes* : https://arxiv.org/abs/1901.01849

[5] Plouffe, Simon, Une formule pour les nombres premiers : http://vixra.org/abs/1902.0036

[6] Debasish Chakraborty1, Snehasish Kar, and Kalyan Guchait, *Efficient Data Compression Using Prime Numbers.*

[7] Math Mahoney, https://encode.ru/threads/1723-Compressing-prime-numbers, Compressing prime numbers.

[8] Dylan Fridman, Juli Garbulsky, Bruno Glecer, James Grime, Massi Tron Florentin, https://arxiv.org/abs/2010.15882