Suite multiplicative

Année 2017-2018

Auteurs : Sevcan LEKESIZ, Medi OLIVIER (1e S) ; François DEGUINE, Joseph TOUZET (terminale S)

Établissement : Lycée Rive Gauche, Toulouse

Encadrés par : Sébastien LACAM, Elise MONNIER, Jean-Christophe SAN SATURNINO,

Chercheuse : Sabine MERCIER, Institut de Mathématiques de Toulouse, Université de Toulouse 2 Jean Jaurès.

1. Introduction

Notre sujet porte sur les suites de nombres. On prend un entier naturel. On fait le produit de tous les chiffres qui le composent. On répète cette opération avec le résultat du calcul précédent à chaque fois jusqu'à obtenir un chiffre.

Exemple :
Si on prend 326 :

$3 \times 2 \times 6 = 36$ (étape 1)
$3 \times 6 = 18$ (étape 2)
$1 \times 8 = 8$ (étape 3)

Il y a 3 étapes et le nombre final est 8.

Conjecture : Il semblerait que la suite se finisse toujours en moins de 11 étapes.

Nous allons commencer par définir les notions importantes ainsi que des fonctions utiles. Dans un second temps nous allons exposer des propriétés. Par la suite nous allons proposer des reformulations du problème et enfin nous montrerons les programmes réalisés.

2. Définitions

On a d'abord cherché à définir une fonction correspondant à la réalisation d'une étape.

Avant de commencer à regarder comment s'écrit la fonction, on va définir quelques écritures pour faciliter la compréhension :

$\sum$ correspond à la somme, autrement dit $\sum_{n=a}^{b} x_n = x_a + x_{a+1} + \cdots + x_b$

On met en bas la valeur de *n* de départ, en haut le rang jusqu'auquel on continue d'additionner et ensuite la forme de chaque terme de la somme.

$\prod$ correspond au produit, et il a la même fonction que le sigma mais en remplaçant la somme par un produit. Autrement dit $\prod_{n=a}^{b} x_n = x_a \times x_{a+1} \times \ldots \times x_b$.

2.1. Fonction « Longueur »

On définit la fonction qui à un nombre entier associe le nombre de chiffres qui le composent. On l'appelle fonction « *longueur* ».

Cette fonction utilise les fonctions logarithme décimal et partie entière :
- le logarithme d'un nombre est la puissance réelle à laquelle on a élevé 10 pour obtenir ce nombre ;
- la partie entière inférieure d'un nombre correspond au plus grand entier inférieur ou égal à ce nombre (1).

La fonction « *longueur* » consiste à prendre le logarithme de *x*, puis à ne garder que sa partie entière et à ajouter 1 à ce résultat.
On obtient ainsi le nombre de chiffres qui composent le nombre initial.

2.2. Fonction « Chiffre »

La fonction « *chiffre* » consiste à extraire le n-ième chiffre d'un nombre.

Pour cela, on décale la virgule de $n-1$ crans en divisant le nombre par 10^{n-1}, puis on prend la partie entière inférieure du nombre obtenu.
On prend ensuite le modulo par 10 du nombre obtenu, c'est à dire le reste de la division de ce nombre par 10.

2.3. Fonction « Étape »

La fonction *d* que nous avons définie s'écrit comme ceci :

$$d(x) = \prod_{n=0}^{\lfloor \log(x) \rfloor} \lfloor 10^{-n} x \rfloor \bmod 10$$

On fait le produit de chiffres un par un, autant de fois qu'il y a de chiffres dans le nombre.
Ici, n correspond au rang du chiffre extrait.
On utilise « produit » pour $n = 0$ (le premier chiffre) jusqu'au nombre total de chiffres, et on va multiplier le chiffre de rang n avec le chiffre de rang $n+1$ autant de fois qu'il y a de chiffres dans le nombre.

3. Propriétés

Nous avons démontré trois propriétés que nous avons appelées « Décroissance », « Facteurs premiers » et « Majoration ».

3.1. Propriété « Décroissance »

Propriété « Décroissance » : pour tout $x \geqslant 10$, on a $x > d(x)$.

Démonstration :
On peut écrire x sous la forme d'une somme de ses chiffres multipliés par leur puissance de 10 respective, avec le premier chiffre supérieur à 0.
$d(x)$ est le produit des chiffres de x, on cherche donc à démontrer que x est supérieur à $d(x)$ ce qui correspond à vérifier que $x - d(x) > 0$.
$x = \sum_{m=0}^{n} a_m \, 10^m$ avec $a_n > 0$,

$$d(x) = \prod_{m=0}^{n} a^m$$

$$x > d(x) \Leftrightarrow x - d(x) > 0$$
$$x - d(x) = \sum_{m=0}^{n} a_m\, 10^m - \prod_{m=0}^{n} a_m = a_n\left(10^n - \prod_{m=0}^{n-1} a^m\right) + \sum_{m=0}^{n-1} a_m\, 10^m$$

or $10^n > \prod_{m=0}^{n-1} a_m$, $10^n - \prod_{m=0}^{n-1} a_m > 0$ et $a_n > 0$, et $\sum_{m=0}^{n-1} a_m 10^m \geq 0$ donc $x - d(x) > 0$.

3.2. Propriété « Facteurs premiers »

Maintenant, nous allons démontrer la seconde propriété, nommée « Facteurs premiers ».

Propriété « Facteurs » : Si un nombre est divisible par un nombre premier à plus d'un chiffre alors il ne peut pas être obtenu par un produit de chiffres, et n'a donc pas d'antécédent par la fonction d.

Par exemple, 33 est divisible par 11 et 3, (11 fois 3 égale 33) il n'a donc pas d'antécédent par la fonction d. Sinon, on peut écrire 11 en produit de chiffres, contradictoire au fait qu'il soit premier. Autrement dit, dire qu'il existe un x tel que $y = d(x)$ revient à dire que y est égal à 0 ou qu'il existe a, b, c et d, entiers naturels, tel que $y = 2^a * 3^b * 5^c * 7^d$.

3.3. Propriété « Majoration »

Soit f une fonction croissante qui majore la fonction d et qui lui est strictement supérieure. Une telle fonction existe car $f(x) = x$ majore strictement d par la propriété de décroissance (2). Alors notons $m(x)$ le nombre d'étapes associé à x, et soit une fonction g telle que f appliquée $g(x)$ fois à x est strictement inférieure à 10. Alors d appliquée $g(x)$ fois à x sera aussi inférieure à 10. Ainsi, g majorera m.

Démonstration :
Soit $f: \mathbb{N} \to \mathbb{N}$où f est croissante, $f \geq d$ sur $\mathbb{N}$, soit m la fonction définie sur $\mathbb{N}$ associant à un nombre sa multiplicité (3), soit $g: \mathbb{N} \to \mathbb{N}$ tel que pour tout $x \in \mathbb{N}$, $f^{g(x)}(x) < 10$.
On a :
$$d(x) \leq f(x)$$
d'où $d^2(x) \leq f(d(x)) \leq f^2(x)$ et de proche en proche on obtient
$$d^k(x) \leq f^k(x) \text{ pour tout } k \text{ appartenant à } \mathbb{N}.$$
On obtient alors :
$$f^{g(x)}(x) < 10 \text{ et } d^{g(x)}(x) < 10$$
et, par définition de $m(x)$, on en déduit que $g(x) \geq m(x)$.

Dans ces conditions, une majoration de g donne une majoration de m.

3.4. Remarques concernant la propriété « Majoration »

g ne peut pas être majorée si $f(\mathbb{N}) = \mathbb{N}$.

Démonstration :
$\forall x \in \mathbb{N}, g(x) \leqslant n$ (4) où $n \in \mathbb{N}, et\ f(\mathbb{N}) = \mathbb{N}$.
Soit $m \in \mathbb{N}$ tel que $g(m) = n$.
Alors il existe $k \in \mathbb{N}$ tel que $f(k) = m$ d'où $g(k) \geq g(m) + 1$, $g(k) \geq n + 1$, ce qui est absurde.

Autrement dit, dans nos conditions, $[f(\mathbb{N}) = \mathbb{N}] \Rightarrow [\nexists n \in \mathbb{N}$ tel que $\forall x \in \mathbb{N}, g(x) \leq n]$.

Remarque :

On aurait pu de même démontrer que

$$[f(\mathbb{N}) \cap \{x|g(x) = n\} \neq \emptyset] \Rightarrow [\nexists n \in \mathbb{N} \text{ tel que } \forall x \in \mathbb{N}, g(x) \leq n] \text{ (5)}$$

4. **Reformulations**

Nous proposons aussi deux reformulations du problème que nous avons nommées « Chiffres premiers » et « Ensembles », que nous allons vous présenter.

4.1. Reformulation « Chiffres premiers »

Un nombre qui serait divisible par un nombre premier possédant au moins deux chiffres ne pourra pas s'écrire sous la forme d'un produit de chiffres et ne pourra donc pas avoir d'antécédent par la fonction d car cette fonction génère un produit de chiffres. Autrement dit,

Reformulation : Un nombre n'a pas d'antécédent par la fonction d si et seulement si il est divisible par un nombre premier supérieur ou égal à 11 (6).

4.2. Reformulation « Ensembles »

On définit C_0 l'ensemble de tous les chiffres et donc l'ensemble de tous les nombres qui finissent en zéro étape. On définit C_1 comme l'ensemble des nombres qui ont pour résultat par la fonction d un chiffre et donc qui finissent en une étape (en excluant C_0) . On peut par la suite définir que l'ensemble C_{m+1} est l'ensemble des nombres donnant un élément de C_m pour tout m supérieur ou égal à 1. L'ensemble C_m est donc l'ensemble des nombres finissant en m étapes. Tous les ensembles s'excluent car tout nombre a une multiplicité unique.

Dire qu'il existe un nombre d'étapes maximal n revient à dire qu'aucun nombre ne finit en $n + 1$ étapes donc que C_{n+1} est vide. Sachant que C_{n+1} est l'ensemble des antécédents de C_n alors aucun élément de C_n ne peut avoir d'antécédent, ce qui selon la propriété « Facteurs premiers » revient à dire que tous les éléments de C_n sont divisibles par un nombre premier supérieur ou égal à 11.

Dire qu'il existe n, nombre d'étapes maximum, implique que pour tout élément x de C_n il existe un nombre premier supérieur ou égal à 11 qui divise x.

5. **Programmes réalisés**

Nous avons donc réalisé un ensemble de programmes pour calculer le nombre d'étapes pour un nombre donné, ceci pouvant nous permettre de vérifier ou d'invalider nos hypothèses.
Le premier programme en java réalisait le produit des chiffres en prenant un nombre comme une chaîne de caractères ou chaque caractère est un chiffre.
Mais cela n'est pas très efficace en temps de calcul.

On a donc eu besoin de réaliser une fonction mathématique réalisant une étape, ce qui a motivé l'écriture de $d(x)$.
Nous avons utilisé Java (environ 20 lignes) pour réaliser la fonction $d(x)$ par facilité car c'est le langage que l'on maîtrisait le mieux.
Nous avons programmé à nouveau la fonction en Python pour mieux communiquer avec l'une de nos encadrante.
On a aussi utilisé le C++ dans un souci de performance, et de passage possible en CUDA, pour tester un plus grand ensemble de nombres.
Par exemple pour l'ensemble de 10 à 1 milliard, en Java le temps de calcul était de 1281 secondes, en Python de 6016 secondes et en C++ de 1847 secondes (programme non parallélisé).

Notre objectif aurait été de trouver par une recherche systématique le premier nombre finissant en 11 étapes (finalement trouvé mais pas par une recherche systématique).
La réalisation et l'amélioration de ces programmes ne devant pas empiéter sur le temps consacré à la recherche mathématique et la mise en forme de notre problème pour le congrès, nous n'avons pas pu réaliser toutes les optimisations possibles. Par exemple ignorer tous les nombres qui auraient le même résultat que des nombres déjà testés comme : tout nombre contenant 0, a pour résultat 0 en 1 étape ; tout nombre contenant un 5 et un nombre pair, finissant par 0 en 2 étapes ; tout nombre résultant d'une permutation des chiffres d'un nombre déjà testé ; tout nombre contenant au moins un 1. Mais aussi la parallélisation des calculs (réaliser plusieurs calculs en même temps) en CUDA pour qu'une carte graphique réalise les calculs...

6. Conclusion

Pour conclure, le problème sur lequel nous avons mené de multiples recherches est un problème qui n'est pas actuellement résolu. Après avoir lu l'article « pour la science » (2013) de Jean-Paul Delahaye (7) nous avons pu ajouter par rapport à cet article que la formulation mathématiques se fait par le biais d'une fonction d. Nous proposons une démonstration différente de la propriété de décroissance. De plus nous avons reformulé le problème.

7. Annexes

Code en Python :

```python
from math import ceil, floor, log10

def d(x):
    y = 1
    for n in range(1+(int) (floor(log10(x)))):
        y *= floor((10**-n)*x) % 10
    return y

def s(x):
    etape = 0
    while x>=10:
            x = d(x)
            etape += 1
    return etape,x

```

Tests avec 77, le nombre inférieur à 100 avec la plus grande persistance multiplicative (8) :

```
>>> resultat = s(77)
>>> print("77 finit en " + str(resultat[0]) + " etapes par " + str(resultat[1]))
77 finit en 4 etapes par 8.0
```

Notes d'édition

(1) La partie entière d'un réel x sera notée $\lfloor x \rfloor$.

(2) Avec $f(x) = x$, on n'a pas la majoration stricte $f(x) > d(x)$ pour tout x, puisque $d(x) = x$ pour $x < 10$ (la propriété de décroissance ne s'applique que pour $x \geq 10$). Plus généralement, on ne peut pas avoir en même temps f croissante, $f(1) > 1$ et $f(\mathbb{N}) = \mathbb{N}$. Mais il suffit pour la suite de supposer la majoration au sens large.

(3) Le terme *multiplicité* peut avoir d'autres sens ; il désigne ici le nombre d'étapes pour arriver à un produit à un chiffre.

(4) Plus précisément, on suppose que g est majorée. Alors, comme elle est à valeurs entières, elle atteint une plus grande valeur et n désigne cette plus grande valeur (on a besoin de cela pour la démonstration qui suit).

(5) La démonstration permet seulement de montrer que si $g(x) = n$ et que x est une valeur de f, alors n ne peut pas être un majorant de g.

(6) Il a seulement été montré que si un nombre a un diviseur premier ≥ 11 alors il n'a pas d'antécédent, alors que cette reformulation affirme que la condition est nécessaire et suffisante (*si et seulement si*) ; mais la réciproque ne pose pas de problème : si un nombre n n'a que des facteurs premiers < 10, il a pour antécédent un nombre dont les chiffres en écriture décimale sont ses facteurs premiers figurant chacun un nombre de fois égal à l'exposant du nombre premier dans la factorisation de n ; par exemple $504 = 2^3 \times 3^2 \times 7$ a pour antécédent 222337 (et tous les nombres obtenus par permutation de ces chiffres).

(7) Cet article, « *La persistance des nombres* », paru dans *Pour la Science* n°430 (août 2013), est accessible en ligne sur la page web de Jean-Paul Delahaye :
http://www.lifl.fr/~jdelahay/pls/2013/237.pdf

(8) La *persistance multiplicative* d'un entier n, appelée ici *multiplicité*, est le nombre d'étapes pour arriver à un nombre à un chiffre à partir de n par multiplications successives des chiffres (voir un historique et des références dans l'article de J.-P. Delahaye).