BROUILLON - SOMMER LES CARRÉS DES CHIFFRES D'UN NATUREL

CHRISTOPHE BAL

Table des matières

1. Faire une tête au carré à tous les entiers naturels

Voici un procédé facile à faire à l'aide d'une calculatrice. Considérons un entier naturel n, puis calculons la somme de ses chiffres élevés au carré. Ceci nous donne un nouveau naturel auquel on peut appliquer le même procédé. Voici deux exemples.

Exemple 1. *Pour $n = 19$, nous obtenons :*

- $1^2 + 9^2 = 82$
- $8^2 + 2^2 = 68$
- $6^2 + 8^2 = 100$
- $1^2 + 0^2 + 0^2 = 1 \rightarrow$ *Rien de nouveau à attendre.*

Exemple 2. *Pour $n = 1\,234\,567\,890$, après $1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 + 7^2 + 8^2 + 9^2 + 0^2 = 285$ nous obtenons :*

- $2^2 + 8^2 + 5^2 = 93$
- $9^2 + 3^2 = 90$
- $9^2 + 0^2 = 81$
- $8^2 + 1^2 = 65$
- $6^2 + 5^2 = 61$
- $6^2 + 1^2 = 37$
- $3^2 + 7^2 = 58$
- $5^2 + 8^2 = 89$
- $8^2 + 9^2 = 145$
- $1^2 + 4^2 + 5^2 = 42$
- $4^2 + 2^2 = 20$
- $2^2 + 0^2 = 4$
- $4^2 = 16$
- $1^2 + 6^2 = 37 \rightarrow$ *Dèjà rencontré.*

Dans le 1er cas, au bout d'un moment le procédé ne produit que des 1. Ce sera le cas dès que l'on commence avec une puissance de 10. Le 2e exemple montre que le mieux que l'on puisse espérer c'est que le procédé devienne périodique à partir d'un moment *(on parle de phénomène ultimement périodique).*

On peut explorer le comportement de ce procédé sur plusieurs valeurs grâce à un programme. Voici un code possible écrit en Python qui prend un peu de temps pour vérifier que pour tous les naturels $n \in [\![1; 10^6]\!]$, le procédé devient ultimement périodique.

Date: 6 Juin 2018 – 8 Déc. 2018.

```
NMAX    = 10**6
MAXLOOP = 10**20

for n in range(1, NMAX + 1):
    nbloops = 0
    results = []

    while nbloops < MAXLOOP and n not in results:
        nbloops += 1
        results.append(n)
        n = sum(int(d)**2 for d in str(n))

    if n not in results:
        print(f"Test raté pour n = {n}.")

print(f"Tests finis.")
```

Il reste à voir ce qu'il se passe dans le cas général. La section qui suit démontre que pour tout naturel n, le procédé sera toujours ultimement périodique.

2. Une preuve

Pour un naturel $n = [c_{d-1}c_{d-2}\cdots c_1c_0]_{10} \overset{\text{def}}{=} \sum_{k=0}^{d-1} c_k 10^k$ avec $c_{d-1} \neq 0$, on pose $sq(n) = \sum_{k=0}^{d-1}(c_k)^2$ et $\texttt{taille}(n) = d$ qui sera appelé *« taille de n »*.

Pour $(n\,;k) \in \mathbb{N}^2$, on définit $\boxed{n}_{\,0} = n$ et $\boxed{n}_{\,k} = sq^k(n) \overset{\text{def}}{=} sq \circ sq \circ \cdots \circ sq(n)$ avec $(k-1)$ compositions si $k > 0$. Autrement dit, $\boxed{n}_{\,k+1} = sq\left(\boxed{n}_{\,k}\right)$.

On note enfin $V_n = \left\{\boxed{n}_{\,k} \mid k \in \mathbb{N}\right\}$ l'ensemble des valeurs prises par la suite $\left(\boxed{n}_{\,k}\right)_k$.

Fait 1. *$\forall n \in \mathbb{N}$, $sq(n) \leqslant 81d$ où $d = \texttt{taille}(n)$.*

Preuve. *Si $n = [c_{d-1}c_{d-2}\cdots c_1c_0]_{10}$ alors $sq(n) = \sum_{k=0}^{d-1}(c_k)^2 \leqslant \sum_{k=0}^{d-1} 9^2 = 81d$.*

Fait 2. *$\forall n \in \mathbb{N}$, notant $d = \texttt{taille}(n)$, nous avons les résultats suivants :*

(1) Si $d \geqslant 4$ alors $\texttt{taille}(sq(n)) < \texttt{taille}(n)$.
(2) Si $d \leqslant 3$ alors $\texttt{taille}(sq(n)) \leqslant 3$.

Preuve. *Notons que $n \geqslant 10^{d-1}$. Le comportement des fonctions 10^{x-1} et $81x$ sur $\mathbb{R}_+^*$ assure l'existence d'un naturel D tel que $\forall d \in \mathbb{N}$, $d \geqslant D$ implique $10^{d-1} > 81d \geqslant sq(n)$. On a même beaucoup mieux : si $10^{D-1} > 81D \geqslant sq(n)$ alors $d \geqslant D$ implique $10^{d-1} > 81d \geqslant sq(n)$.*

Comme $10^3 > 81 \times 4$, nous avons sans effort le 1er point (rappelons que $10^k > n$ implique que n admet au plus $(k-1)$ chiffres).

Pour $d \leqslant 3$, le 2nd point découle de $sq(999) = 243$, $sq(99) = 162$ et $sq(9) = 81$.

Fait 3. *$\forall n \in \mathbb{N}$, l'ensemble V_n est fini et donc la suite $\left(\boxed{n}_{\,k}\right)_{k\in\mathbb{N}}$ est ultimement périodique, i.e. périodique à partir d'un certain rang.*

Preuve. *Le 2nd point dépend directement du 1er point via le principe des tiroirs et la définition récursive de la suite $\left(\boxed{n}_{\,k}\right)_k$.*

Pour le 1er point, il suffit de montrer que $V_n \subset \left[0; 10^{\texttt{taille}(n)}\right]$ pour $n \geqslant 4$ via une petite récurrence descendante finie, et pour $n \leqslant 3$ on a directement $V_n \subset [0; 10^3]$.

3. Code : déterminer la période d'un naturel

Quand il ne se fige pas, le code suivant donne la *« période »* d'un naturel auquel on applique le procédé.

```
n = 19

results = []
print(n)

while n not in results:
    results.append(n)
    n = sum(int(d)**2 for d in str(n))

print("Période :")
print(results[results.index(n):])
```