
D'après le sujet 2016

Somme de puissances entières d'entiers

1. Somme, somme de carrés, somme de cubes

Les ensembles $A = \{1, 9, 11\}$ et $B = \{3, 5, 13\}$ possèdent des propriétés qui attisent la curiosité.

On remarque en effet que $1 + 9 + 11 = 3 + 5 + 13$ et que $1^2 + 9^2 + 11^2 = 3^2 + 5^2 + 13^2$.

Dans la suite, on dira que la paire d'ensembles d'entiers $\{A, B\}$ possède la propriété S_1 si les deux ensembles ont le même nombre d'éléments et si la somme des éléments de A est égale à la somme des éléments de B . On dira qu'elle possède la propriété S_2 si elle possède la propriété S_1 et si de plus la somme des carrés des éléments de A est égale à la somme des carrés des éléments de B . On dira qu'elle possède la propriété S_3 si de plus la somme des cubes des éléments de A est égale à la somme des cubes des éléments de B .

Étant donné un entier impair n , on se demande s'il est possible de partager l'ensemble $\{0, 1, 2, 3, \dots, n-1, n\}$ en deux parties A et B telles que $\{A, B\}$ possède une des propriétés évoquées ci-dessus. Par convention, A sera la partie qui contient 0.

1. Dans le cas $n = 3$, peut-on trouver $\{A, B\}$ possédant la propriété S_1 ?
2. Même question dans le cas $n = 5$.
3. a) Si l'ensemble $\{0, 1, 2, 3, 4, 5, 6, 7\}$ peut être partagé en deux parties A et B telles que les propriétés S_1 et S_2 soient satisfaites, quelle est la somme des éléments de A ?
b) Quelle est la somme des carrés des éléments de A ?
c) Déterminer des parties A et B solutions du problème.

4. Lorsque l'ensemble $\{0, 1, 2, 3, \dots, n-1, n\}$ peut être partagé de sorte à satisfaire une ou plusieurs des propriétés S_i , on adopte le système de représentation suivant : chaque case de la seconde ligne des tableaux ci-dessous contient 1 si le nombre appartient à A, 0 si le nombre appartient à B.

a) Compléter les deux tableaux :

$n = 3$

0	1	2	3
1			

$n = 7$

0	1	2	3	4	5	6	7
1							

b) L'observation de ces deux tableaux fait naître l'idée qu'en dédoublant chaque 0 en 01 et chaque 1 en 10, on transforme une séquence intéressante en une autre, dont la taille est doublée.

Compléter le tableau correspondant à $n = 15$ et vérifier que les parties obtenues possèdent bien la propriété S_3 .

2. Naissance d'une suite

Inspirés par les questions précédentes, on étudie la suite définie par $t_0 = 1$ et la relation de récurrence :

pour tout entier n , $t_{2n} = t_n$ et $t_{2n+1} = 1 - t_n$

(suite de Prouhet-Thue-Morse).

1. Calculer t_{2018} .
2. Écrire un algorithme permettant, un entier n étant donné, d'obtenir la valeur de t_n .
3. La suite (t_n) possède-t-elle trois termes consécutifs identiques ?
4. Cette suite est-elle périodique ?

3. L-system

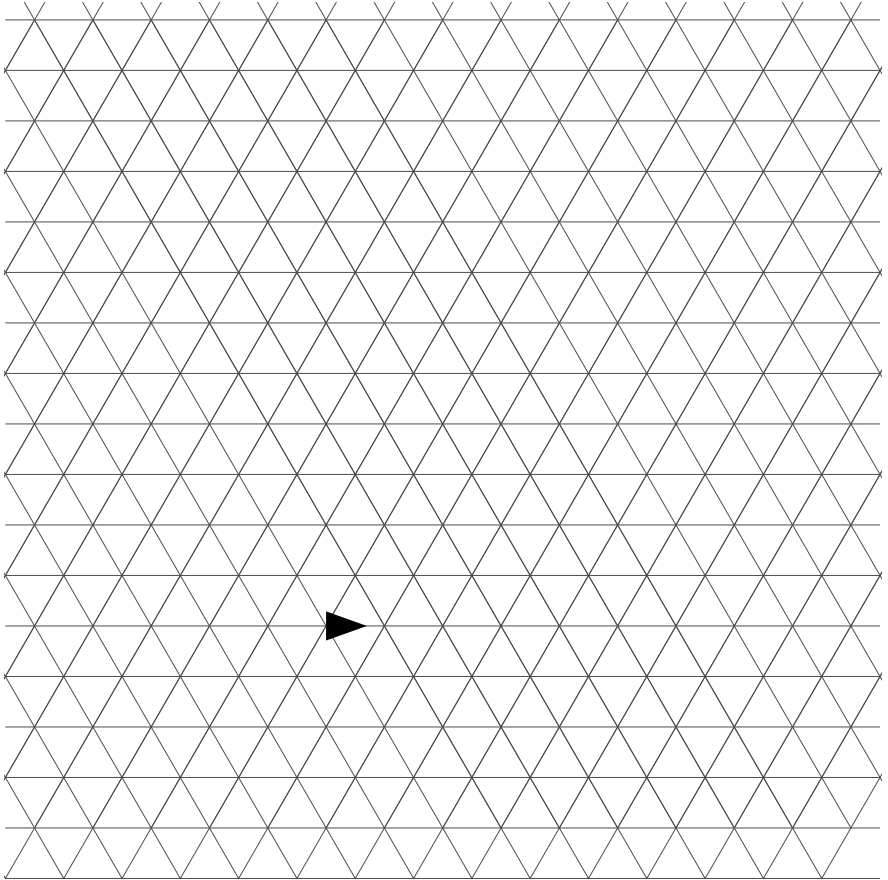
Un *L-system* ou un *système de Lindenmayer* est un système de grammaire formelle, inventé en 1968 par le biologiste hongrois Aristid LINDENMAYER. Un L-system modélise le processus de développement et de prolifération de plantes ou de bactéries.

La page de Wikipédia en français pour comprendre l'idée : <https://fr.wikipedia.org/wiki/L-Système>, celle en anglais, plus complète : <https://en.wikipedia.org/wiki/L-system> et le livre *The Algorithm Beauty of Plants*, Przemyslaw PRUSINKIEWICZ et Aristid LINDENMAYER <http://algorithmicbotany.org/papers/abop/abop.pdf> (attention ≈ 16 Mio).

L'idée est d'associer la *tortue LOGO* à la *chaîne de caractères* associée à t_n , le terme de rang n de la suite définie précédemment. Ici les caractères utilisés sont 0 et de 1.

À chaque caractère on associe un mouvement de la tortue : 0 signifie « avancer de 1 unité » et 1 « tourner à gauche de 60° (sans avancer) ».

1. À l'aide d'un programme (ou de courage), obtenir tous les termes de la suite (t_n) jusqu'à t_{127} .
2. Dessiner alors le parcours de la tortue en utilisant le quadrillage (la tortue est orientée vers l'est).



3. Après l'avoir compris, compléter l'algorithme, puis le taper.

```
1 #-*- coding: utf-8 -*-
2 # pyhton3
3 #
4 # d'après Olympiades de Mathématiques
5 # Amérique - Antilles - Guyane 2016
6 # et Wikipédia en anglais : Thue-Morse sequence
7
8 import turtle as tl
9
10 # création de la liste sous forme de chaîne
```

```

11 L="1001"
12 for i in range(6): # répéter 6 fois
13     tmp=""
14     for c in L: # pour chaque caractère <c> de la chaî
        ↪ne <L>
15         if c== "0" :
16             tmp=tmp+"01"
17         else:
18             tmp=tmp+"10"
19     L=tmp[:]
20
21 # préparation de la tortue
22 t1.penup()
23 t1.setposition(-300,-300)
24 t1.pendown()
25 t1.hideturtle()
26 # dessin fait par la tortue lors du parcours de la cha
        ↪îne
27 for c in L: # pour chaque caractère <c> de la cha
        ↪îne <L>
28     if c== "0":
29         t1.forward(100)
30     else:
31         t1.backward(100)
32
33 t1.mainloop()

```

pour finir (ou commencer une nouvelle aventure) et jouer avec les L-system :
<http://www.kevs3d.co.uk/dev/lsystems>