

Sommons les diviseurs

Année 2025 – 2026

AMARA Anaïs et LOPACZYK Emmy en classe de troisième

DERVEAUX Faustine et ZAMORA Lucie en classe de seconde

Établissements : le Collège A. MALRAUX de LAMBRES-LEZ-DOUAI, et le Lycée A. RIMBAUD de SIN-LE-NOBLE

Enseignantes : BOCQUET Amélie, PICAVEZ Marion

Chercheur : François GOICHOT, Ceramaths - UPHF

1. Présentation du sujet

Pour un entier $n \geq 1$, on note $s(n)$ la somme de ses diviseurs propres, c'est-à-dire autres que n lui-même. Quelles propriétés peut-on trouver à $s(n)$ en général ?

Par exemple :

- Quand est-ce que $s(n) = 1$?
- Y a-t-il des valeurs de n pour lesquelles on peut trouver $s(n)$ sans calcul ?
- Que se passe-t-il si on répète l'opération, c'est-à-dire si on considère $s(s(n))$, $s(s(s(n)))$, etc.
- Est-il vrai que pour tout $m \geq 1$, on peut trouver n tel que $s(n) = m$?

1.1. Exemple :

Pour $n = 25$, on a $5 * 5$ et $1 * 25$.

Alors les diviseurs de 25 (autres que lui-même) sont 1 et 5.

Donc $s(25) = 1 + 5 = 6$.

1.2. Nos programmes :

Nous avons réussi à calculer à la main $s(n)$ pour les 50 premiers entiers mais afin de travailler avec des nombres plus grands, ou généraliser avec plus d'exemples, ou aller plus vite dans nos recherches, nous avons créé un programme en langage Scratch pour les collégiens et en langage Python pour les lycéens.

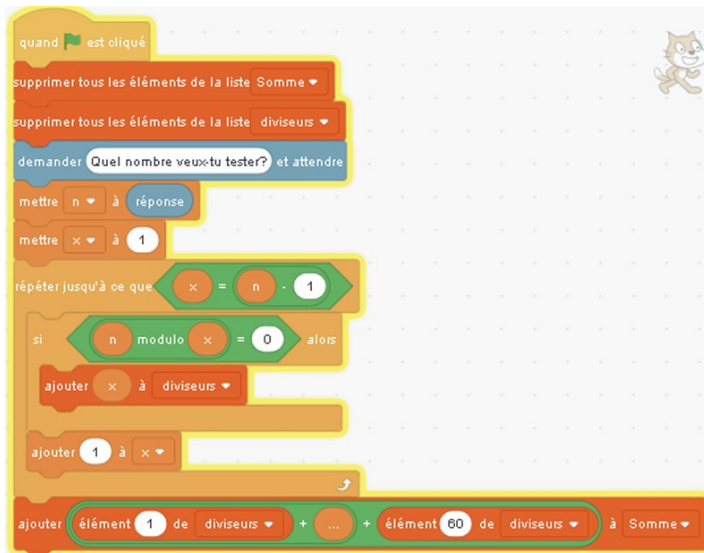


Figure 1 Programme Scratch

```
def somme_diviseurs(n):
    s=0
    for i in range(2,n+1):
        if n % i == 0 : # i divise n
            q = n // i
            s = s + q
    return s
```

Figure 2 Programme Python

2. Pour quelle(s) valeur(s) de n a-t-on $s(n) = 1$?

Nombre premier :

Rappel : Un nombre premier est un nombre entier supérieur ou égal à deux divisible uniquement par lui-même et par 1.

Exemple : 71 est un nombre premier car $71 = 1 * 71$ et il n'y a pas d'autres possibilités.

Alors $s(71) = 1$.

Donc $s(n) = 1$ est vrai lorsque n est un nombre premier et inversement.

3. Trouver $s(n)$ sans calcul

3.1. Quand n est un nombre premier

Exemples : 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, ...

3.2. Quand n est une puissance de 2,

$$s(2^m) = 2^m - 1$$

Exemple : $n = 8 (= 2^3)$ On a $8 = 4 * 2 = 1 * 8$. (1)

Ainsi ses diviseurs sont 1 ; 2 ; 4 et 8 que l'on ne prend pas.

Donc $s(8) = 1 + 2 + 4 = 7$. (2)

3.3. Quand n est un carré d'un nombre premier,

$s(p^2) = p + 1$ avec p nombre premier. (3)

Exemple : $n = 9 (= 3^2)$ On a $9 = 3 * 3 = 1 * 9$.

Alors ses diviseurs sont 1 ; 3 et 9 que l'on ne prend pas.

Donc $s(9) = 1 + 3 = 4$.

3.4. Quand n est le produit de 2 nombres premiers différents,

$s(p_1 * p_2) = p_1 + p_2 + 1$ avec p_1 et p_2 deux nombres premiers différents. (4)

Exemple : $n = 213 = 3 * 71$. On a bien 3 et 71 deux nombres premiers différents.

Alors les diviseurs de 213 sont alors 1 ; 3 ; 71 et 213 que l'on ne prend pas.

Donc $s(213) = 1 + 3 + 71 = 75$.

4. Les nombres parfaits

On appelle nombre parfait un nombre qui vérifie : $s(n) = n$.

Exemple : pour $n = 6$ on a $6 = 3 * 2 = 1 * 6$.

Alors $s(6) = 3 + 2 + 1 = 6$.

On a trouvé grâce à notre programme :

```
def nombre_egaux():  
    liste_nombre_egaux=[]  
    for n in range (1,1001):  
        s = somme_diviseur(n)  
        if s == n:  
            liste_nombre_egaux =liste_nombre_egaux + [n]  
    return liste_nombre_egaux
```

Pour les entiers inférieurs ou égaux à 10 000 :

$s(6) = 6$,

$s(28) = 28$,

$s(496) = 496$,

$s(8128) = 8128$.

5. Les nombres en boucle

Un nombre en boucle est un nombre qui vérifie $s(s(n)) = n$.

Avec le programme Scratch :



Avec le programme Python :

```
def boucle_diviseur():
    liste_boucle_infini=[]
    for n in range (200,350):
        nb_etape=0
        s = somme_diviseurs(n)
        while (s != 1) and (nb_etape<10) and (s!=n):
            s = somme_diviseurs(s)
            nb_etape= nb_etape + 1
        if s == n:
            liste_boucle_infini = liste_boucle_infini + [n]
    return (liste_boucle_infini)
```

Pour les entiers inférieurs ou égaux à 10 000, nous avons trouvé comme nombres en boucles les nombres : 220 ; 284 ; 1184 ; 1210 ; 2620 et 2924. (5)

En effet :

$s(220) = 284$ et $s(284) = 220$,

$s(1184) = 1210$ et $s(1210) = 1184$,

$s(2620) = 2924$ et $s(2924) = 2620$.

6. Répétition de la fonction $s(n)$

Exemple : Si $s(66) = 78$, alors $s(s(66)) = s(78) = 90$.

Exemple de répétition : $s(40) = 50 \rightarrow s(50) = 43 \rightarrow s(43) = 1$.

On remarque qu'en répétant la fonction, on finit quasiment toujours par tomber sur un nombre premier, ce qui donne donc 1, mais on peut aussi tomber sur des nombres parfaits et des nombres « en boucle ».

7. Un nombre parfait des nombres en boucle $s(n) = m$?

Est-ce que pour tout $m \geq 1$, on peut trouver un n tel que $s(n) = m$?

C'est-à-dire : Est-ce que tous les nombres apparaissent en tant que somme de diviseurs ?

La réponse est non car par exemple 2 et 5 ne peuvent pas être le résultat de $s(n)$, en effet :

- 2 ne peut être une somme de diviseurs car pour passer de 1 à 2, il faut ajouter 1 ; or on ne compte qu'une fois 1 ;
- 5 ne peut être une somme de diviseurs car comme on ne peut pas compter deux fois le même nombre, pour passer de 1 à 5, la seule possibilité est $s(n) = 1 + 4$. Or si 4 divise n , alors 2 aussi.

Notes d'édition

(1) On pourra remarquer que les diviseurs propres de 2^3 sont 2^0 , 2^1 et 2^2 .

(2) En suivant cet exemple, on peut en déduire quels sont les diviseurs propres de 2^m pour un entier m strictement positif quelconque ?

(3) On pourra remarquer que pour un nombre p premier quelconque, les diviseurs propres de p^2 sont 1 et p .

(4) On pourra remarquer que pour deux nombres premiers quelconques p_1 et p_2 , les diviseurs propres de $p_1 \cdot p_2$ sont 1, p_1 et p_2 . Notons également que le cas traité en 3.3 est un cas particulier de celui traité ici.

(5) La définition des nombres en boucle inclut les nombres parfaits. Il faudrait donc ajouter à cette liste les nombres parfaits trouvés au paragraphe précédent.