

Devoir maison n°4

Suites

À rendre pour le vendredi 18 décembre.

Ce devoir est à faire à deux personnes maximum (un seul travail rendu pour 2 élèves).

On appelle *suite de Syracuse* une suite (u_n) d'entiers naturels définie de la manière suivante :

- le premier terme u_0 est un entier naturel non nul que l'on pourra choisir;
- pour tout entier $n \geq 0$:

$$\begin{cases} u_{n+1} = \frac{u_n}{2} & \text{si } u_n \text{ est pair} \\ u_{n+1} = 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$$

1. (a) Que remarque-t-on lorsqu'on calcule les premiers termes de la suite de Syracuse en prenant $u_0 = 1$?
(b) Le nombre 1 figure-t-il dans la suite de Syracuse lorsque $u_0 = 10$?

La conjecture de Syracuse, non démontrée à ce jour, s'énonce ainsi : « Quel que soit l'entier naturel choisi pour u_0 , le nombre 1 est atteint par un terme de la suite »

Par ailleurs, l'observation de la suite montre que la suite peut s'élever assez haut avant de retomber. Les graphiques font penser à la chute chaotique d'un grêlon ou bien à la trajectoire d'une feuille emportée par le vent. De cette observation est né tout un vocabulaire imagé : on parle du *vol* de la suite et on définit alors :

le temps de vol : c'est le plus petit indice de n tel que $u_n = 1$;

le temps de vol en altitude : c'est le plus petit indice n tel que $u_{n+1} < u_0$;

l'altitude maximale : c'est la valeur maximale de la suite.

2. L'objectif est de créer un algorithme permettant de déterminer le temps de vol de la suite.
 - (a) Compléter le programme en Python ci-dessous.

```
def Syracuse(u):
    if u % 2 == 0:
```

```
        u = ...
    else:
        u = ...
    return u
```

```
def Liste_Syracuse(u):
    L=[u]
    while u != 1:
        L.append(Syracuse(u))
        u=Syracuse(u)
    return L
```

Remarque. «!=» signifie « différent de » et % fournit le reste de la division euclidienne d'un nombre par un autre.

- (b) Quel nombre est renvoyé par *Syracuse*(6)? *Syracuse*(7)?
 - (c) À quel test correspond $u\%2 == 0$?
 - (d) Quel est le rôle de la fonction *Syracuse*?
3. (a) Que retourne l'instruction : *Liste_Syracuse*(14)?
(b) Que fait l'instruction : *L.append(Syracuse(u))*?
(c) Est-on sûr que pour tout entier naturel u non nul, *Liste_Syracuse(u)* renvoie une liste?
 4. On rappelle que *len(L)* renvoie la longueur de la liste L et *max(L)* renvoie la plus grande valeur de la liste L .
 - (a) Si $u_0 = 7$, quelle est le temps de vol de la suite et quelle est son altitude maximale?
 - (b) Écrire un programme qui détermine la plus petite valeur u_0 qui donne un temps de vol supérieur à 100.
 - (c) Quelle est alors l'altitude de ce vol?
 5. Écrire un algorithme permettant d'obtenir le temps de vol en altitude pour une valeur donnée de u_0 .