

Conjecture de Syracuse

Introduction

On considère l'algorithme ci-dessous où n est un entier naturel avec $n > 1$.

```

1  n entier > 1
2
3  Tant que n > 1
4      Si n pair n ←  $\frac{n}{2}$ 
5      Sinon n ←  $3 \times n + 1$ 
6  Afficher n
    
```

Exemple :

si la valeur de départ de n est 10, on obtient
l'affichage de la séquence de nombres :
5 ; 16 ; 8 ; 4 ; 2 ; 1.

PARTIE A étude préliminaire

1. Inscrire dans le tableau les séquences affichées pour chaque valeur de départ.

Départ	Séquence des valeurs affichées en sortie															T_{vol}	A_{max}
11	34	17	52	26	13	40	20	10	5	16	8	4	2	1			
3																	
12																	
32																	
8																	

2. Que remarque-t-on pour chaque séquence ?
3. Pour certaines valeurs de départ, les séquences seront uniquement constituées de nombres pairs avant d'atteindre 1. Sous quelle forme peut-on écrire les nombres de départ associés à de telles séquences ?
-
4. Quelles sont les séquences uniquement constituées de nombres impairs ?
-

Définition

- * On appelle **temps de vol** de la séquence et on note T_{vol} , le nombre de valeurs affichées en sortie.

Pour notre exemple, si $n = 10$, $T_{vol} = 6$ car 6 valeurs sont affichées.

- * On appelle **altitude maximale** et on note A_{max} la plus grande valeur de la séquence.

Pour $n = 10$, $A_{max} = 16$.

5. Compléter les deux dernières colonnes du tableau T_{vol} et A_{max} .

PARTIE B programmation Python

1. On a saisi sur colaboratory une fonction `syracuse` permettant d'obtenir la séquence associée au choix d'un entier de départ n . Ouvrir le fichier intitulé `syracuse` partagé et vérifier les séquences affichées en appelant `syracuse(11)` et `syracuse(32)`.

2. Finir de commenter le programme après les signes `#` :

```
1 def syracuse(n):
2     while n>1:
3         if n%2==0:      #
4             n = n//2    # division entière par 2
5         else:          #
6             n = 3*n+1   #
7         print(n)
```

3. On souhaite utiliser une fonction `tempsvol` qui prend en paramètre un entier n et qui renvoie le temps de vol de la séquence de premier terme n .

```
1 def tempsvol(n):
2     T=0                  # On initialise le temps de vol à 0
3     while n>1:
4         T= .....        # à chaque passage dans la boucle, on incrémente T
5         if n%2==0:
6             n = n//2
7         else:
8             n = 3*n+1
9     return(T)           # la fonction renvoie la valeur finale de T
```

Compléter la ligne 4 de la fonction `tempsvol` sur le fichier collaboratory et exécuter la cellule.

4. Vérifier que le temps de vol si $n = 27$ vaut 111. (Appeler `tempsvol(27)` et exécuter)

5. On considère le programme ci-dessous :

```
1 for i in range(2,50):      # Boucle pour i allant de 2 à 49
2     print("Pour le nombre de départ ",i," le temps de vol est: ",tempsvol(i))
```

Exécuter ce programme et observer les valeurs obtenues, que peut-on dire du temps de vol associé à 27 ?

.....

6. Modifier le dernier programme pour qu'il affiche les temps de vol pour n allant de 2 à 100 inclus. A-t-on constaté un nouveau record de temps de vol ?

7. Si le nombre de départ n s'écrit sous la forme 2^k où $k \in \mathbb{N}$ avec $k > 1$, quel sera le temps de vol ?

.....