

1. Suite de Syracuse

La suite de Syracuse est définie par $u_0 = a$

$$\text{et pour tout } n \in \mathbb{N} \quad u_{n+1} = \begin{cases} u_n \div 2 & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{sinon} \end{cases}$$

1. Écrire une fonction qui donne les n premiers termes de la suite de Syracuse en fonction de a (choisi par l'utilisateur).
2. Émettre une conjecture sur la limite de cette suite.

2. Vol en altitude

Pour simplifier, on cherche le nombre de termes strictement supérieurs à u_0 parmi les n premiers calculés.

1. Écrire une fonction qui renvoie le nombre de termes strictement supérieurs à u_0 parmi les 30 premiers. L'utilisateur choisi la valeur de u_0 .
2. Remarquer des valeurs particulières de u_0 .
Démontrer que pour ce type de valeurs de u_0 la suite est strictement décroissante jusqu'au premier terme qui vaut 1.



```
1  -*- coding : utf8 -*-
2  # python 3
3
4  from matplotlib import pyplot as plt
5  import numpy as np
6
7  def syracuse(a, n):
8      """ liste des n premiers termes de Syracuse
9      pour u_0 = a
10     """
11     sy = [a]
12     for i in range(n):
13         if sy[i] % 2 == 0:
14             sy = sy + [sy[i] / 2]
15         else:
16             sy = sy + [3 * sy[i] + 1]
17     return sy
18
19 n = 50
20 while True:
21     a = int(input("entier a = "))
22     if a == 0:
23         break
24     else:
25         plt.cla() #efface le graphe courant
26         plt.grid()
27         plt.plot(range(n + 1), syracuse(a, n), 'o-')
28         plt.show()
```

```
1  -*- coding : utf8 -*-
2  # python 3
3
```

```
4 from matplotlib import pyplot as plt
5 import numpy as np
6
7 def syracuse(a, n):
8     """ liste des n premiers termes de Syracuse pour u_0 = a
9     """
10    sy = [a]
11    for i in range(n):
12        if sy[i] % 2 == 0:
13            sy = sy + [sy[i] / 2]
14        else:
15            sy = sy + [3 * sy[i] + 1]
16    return sy
17
18 def nb_sup_a(a, n):
19     """ nombre de valeurs > a parmi les n premiers termes
20     """
21    return len([x for x in syracuse(a, n) if x > a])
22
23 def affiche_nb_sup_a(a_max, n=50):
24    x = range(1, a_max + 1)
25    y = [nb_sup_a(a, n) for a in x]
26    plt.grid()
27    plt.bar(x, y, .2, color='b' )
28    plt.title("nombre de termes > u_0 \n parmi les " + \
29             str(n) + " premiers de la suite")
30    plt.show()
31
32 if __name__ == "__main__":
33    affiche_nb_sup_a(50, n=50)
```
