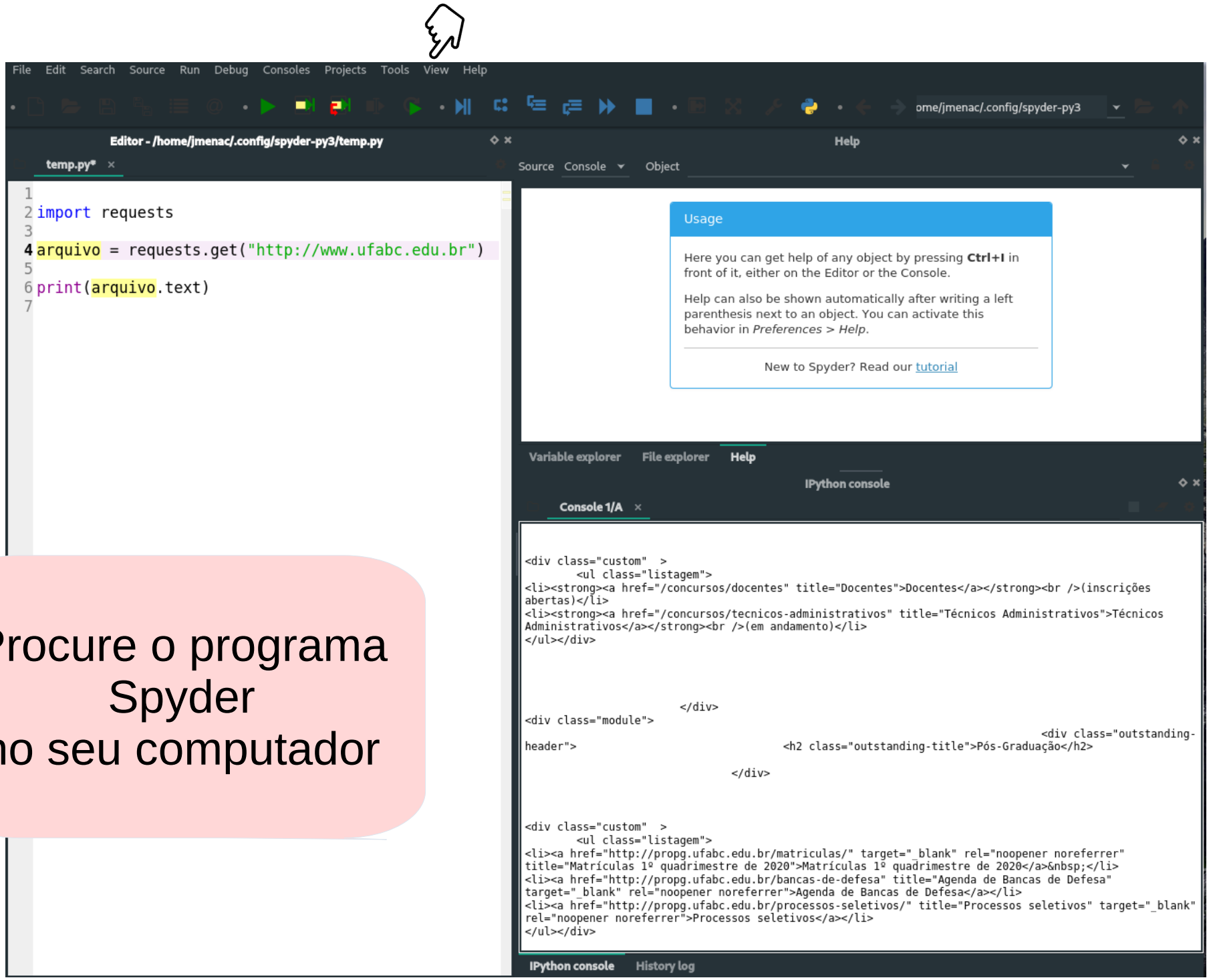
Processamento da Informação

Estruturas de repetição - Parte 2

Prof. Jesús P. Mena-Chalco
CMCC/UFABC

Q1/2020



Procure o programa Spyder
no seu computador

Um Jogo 3n+1

3n+1

Você recebe uma quantidade de dinhero:

- Se o valor for **par**, então dividimos por 2.
- Se o valor for **ímpar**, então multiplicamos por 3 e adicionamos 1.

Você consegue adivinhar se ficaria milhonario(a) dada uma quantia **x** de dinheiro, e aplicar a mesma regra muitas vezes?

3n+1

$$F(n) = \begin{cases} n / 2 & , \text{ se n é par} \\ 3n + 1 & , \text{ se n é ímpar} \end{cases}$$

3n+1

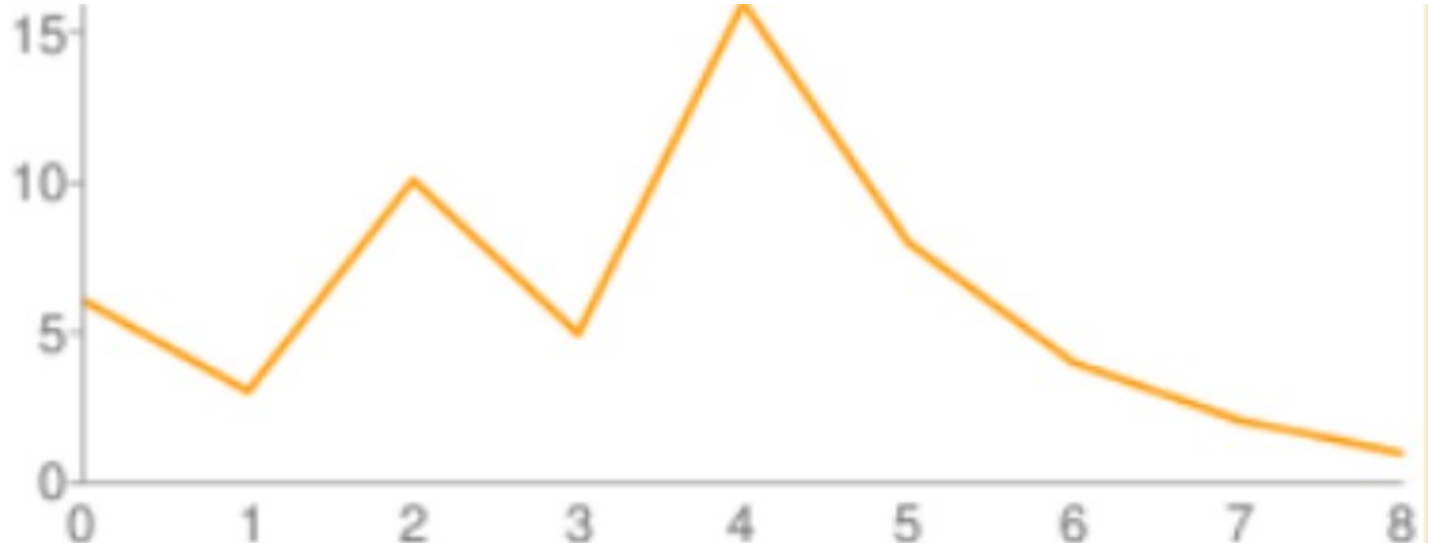
Para n=4 temos:
$4 \rightarrow 2 \rightarrow 1$

Para n=5 temos:
$5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

Para n=6 temos:
$6 \rightarrow 3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

3n+1

Para n=6 temos:
6 → 3 → 10 → 5 → 16 → 8 → 4 → 2 → 1



Número de passos até chegar em 1 = 9

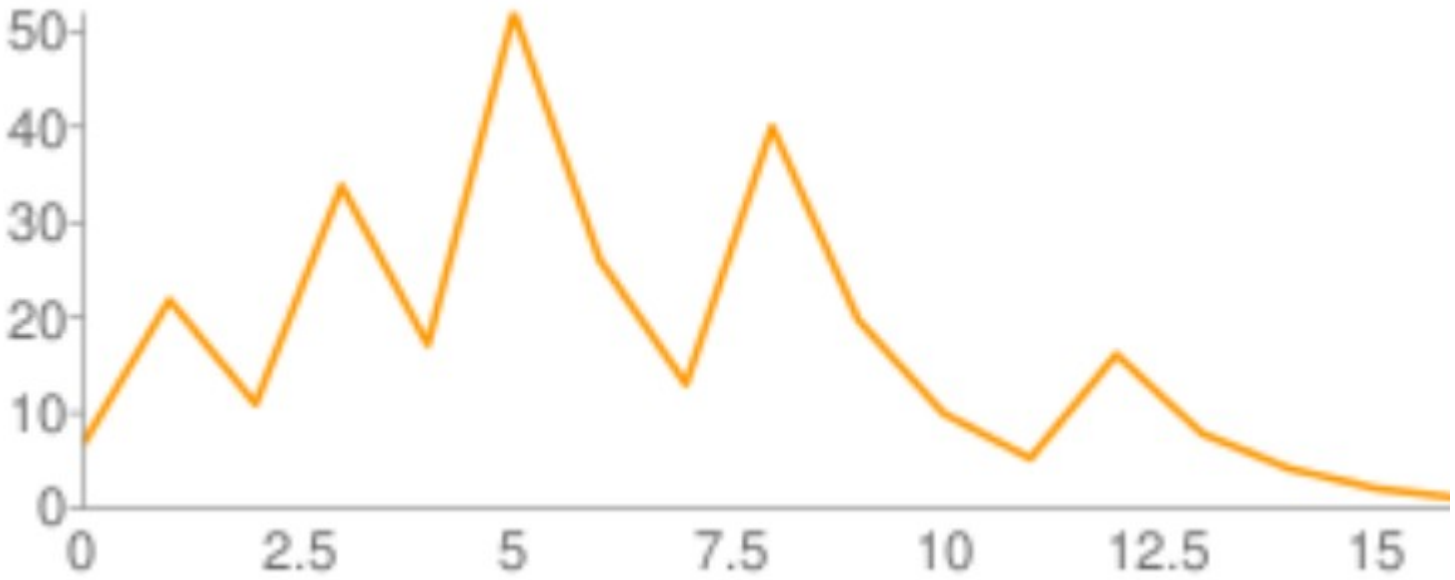
3n+1

Para n=7 ?
Para n=8 ?
Para n=9 ?

3n+1

Para **n=7** (17 passos):

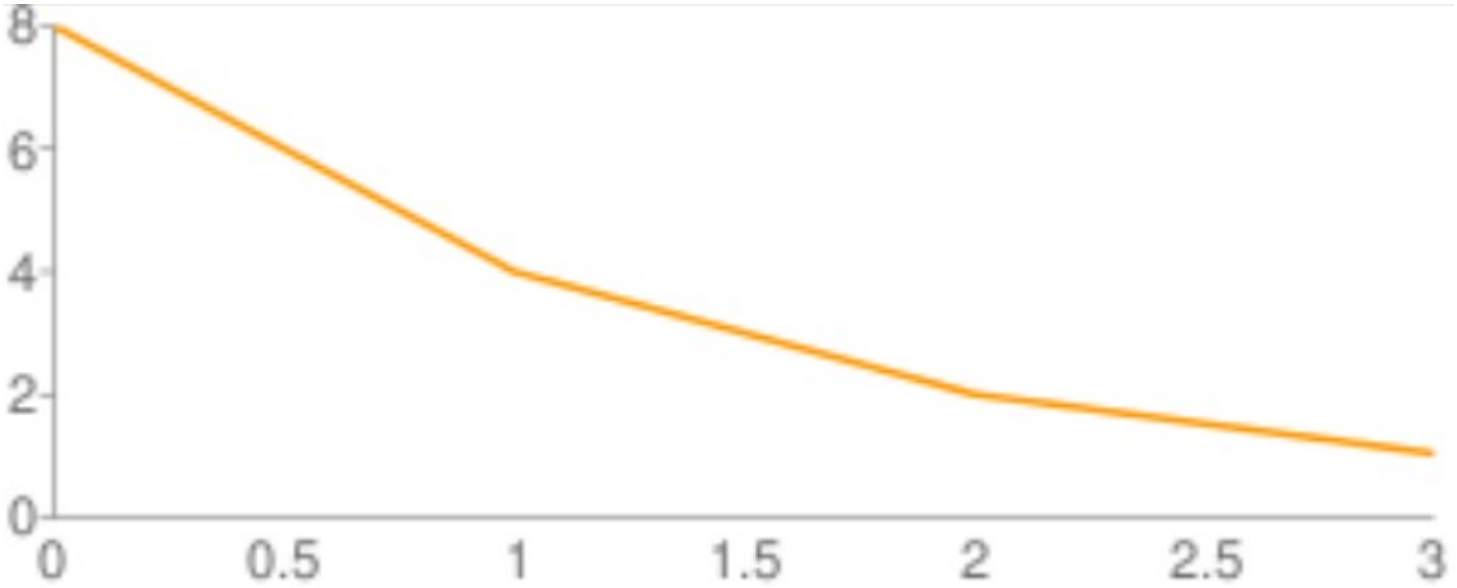
7 → 22 → 11 → 34 → 17 → 52 → 26 → 13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1



3n+1

Para **n=8** (4 passos):

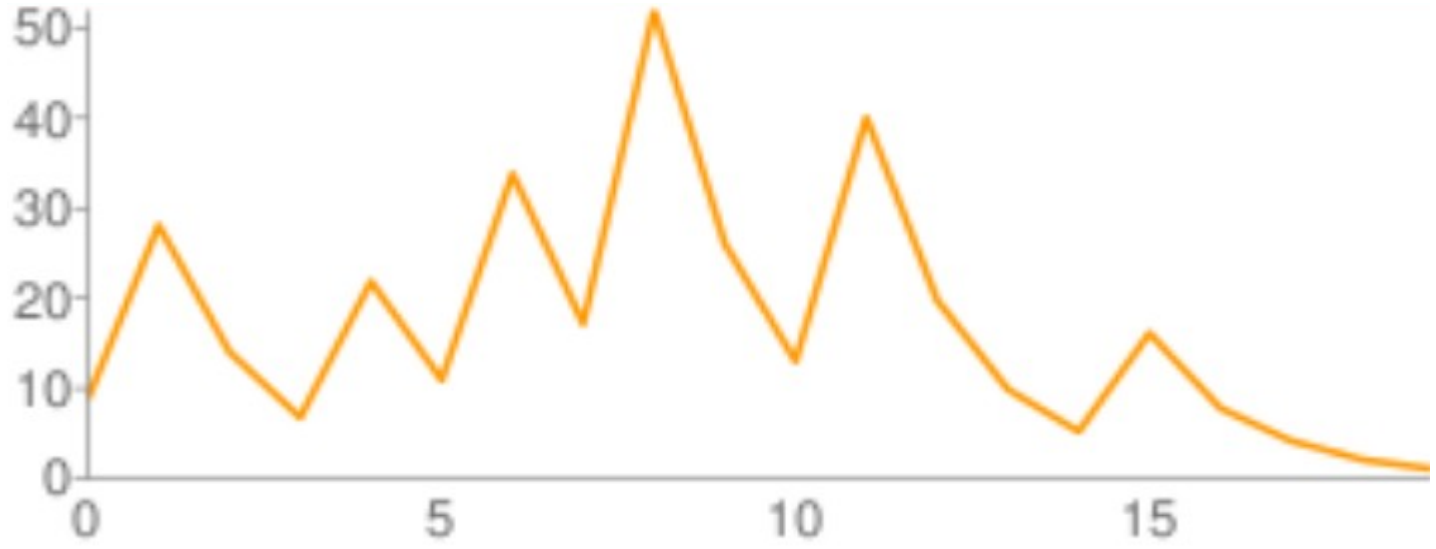
8 → 4 → 2 → 1



3n+1

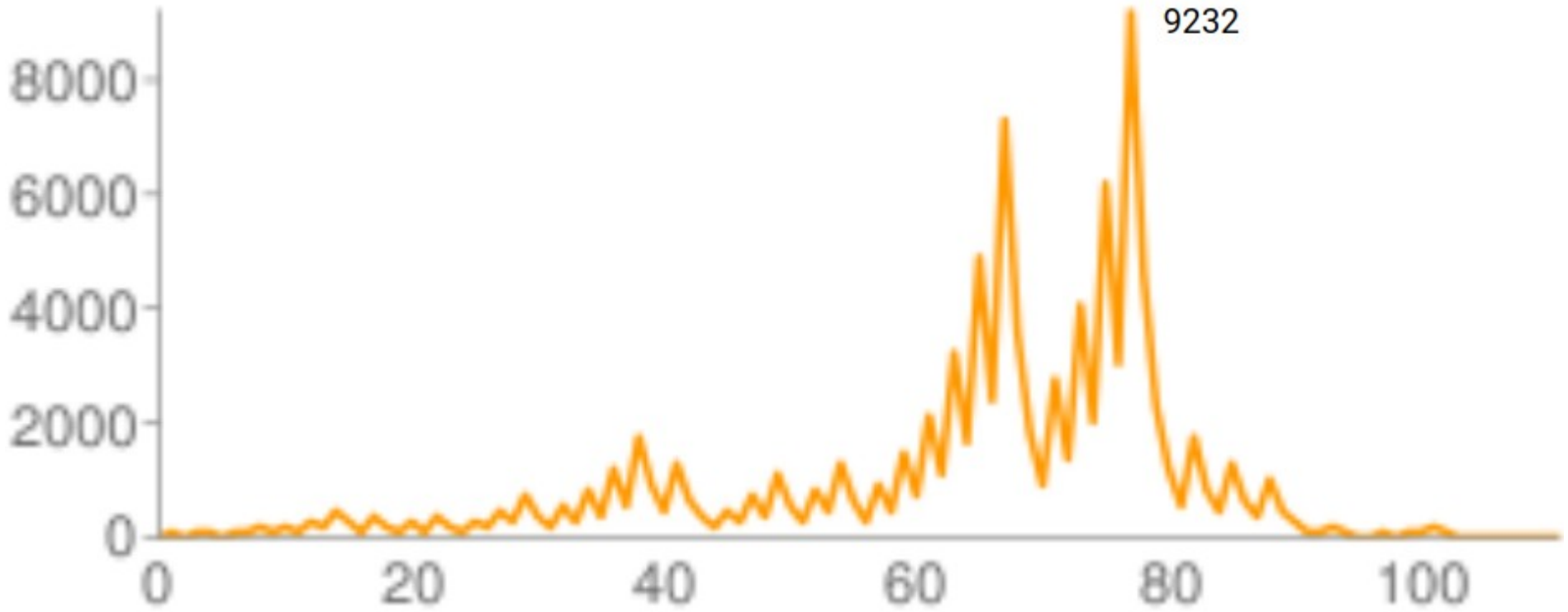
Para **n=9** (20 passos):

9 → 28 → 14 → 7 → 22 → 11 → 34 → 17 → 52 → 26 → 13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1



3n+1

Para **n=27** (112 passos):



3n+1

Crie um programa em que, dado um número inteiro **n**, imprima a sequência de números, segundo o jogo **3n+1**.

$$F(n) = \begin{cases} n / 2 & , \text{ se n é par} \\ 3n + 1 & , \text{ se n é ímpar} \end{cases}$$

Para n=9

```
9
28
14
7
22
11
34
17
52
26
13
40
20
10
5
16
8
4
2
Finalizou!
```

3n+1

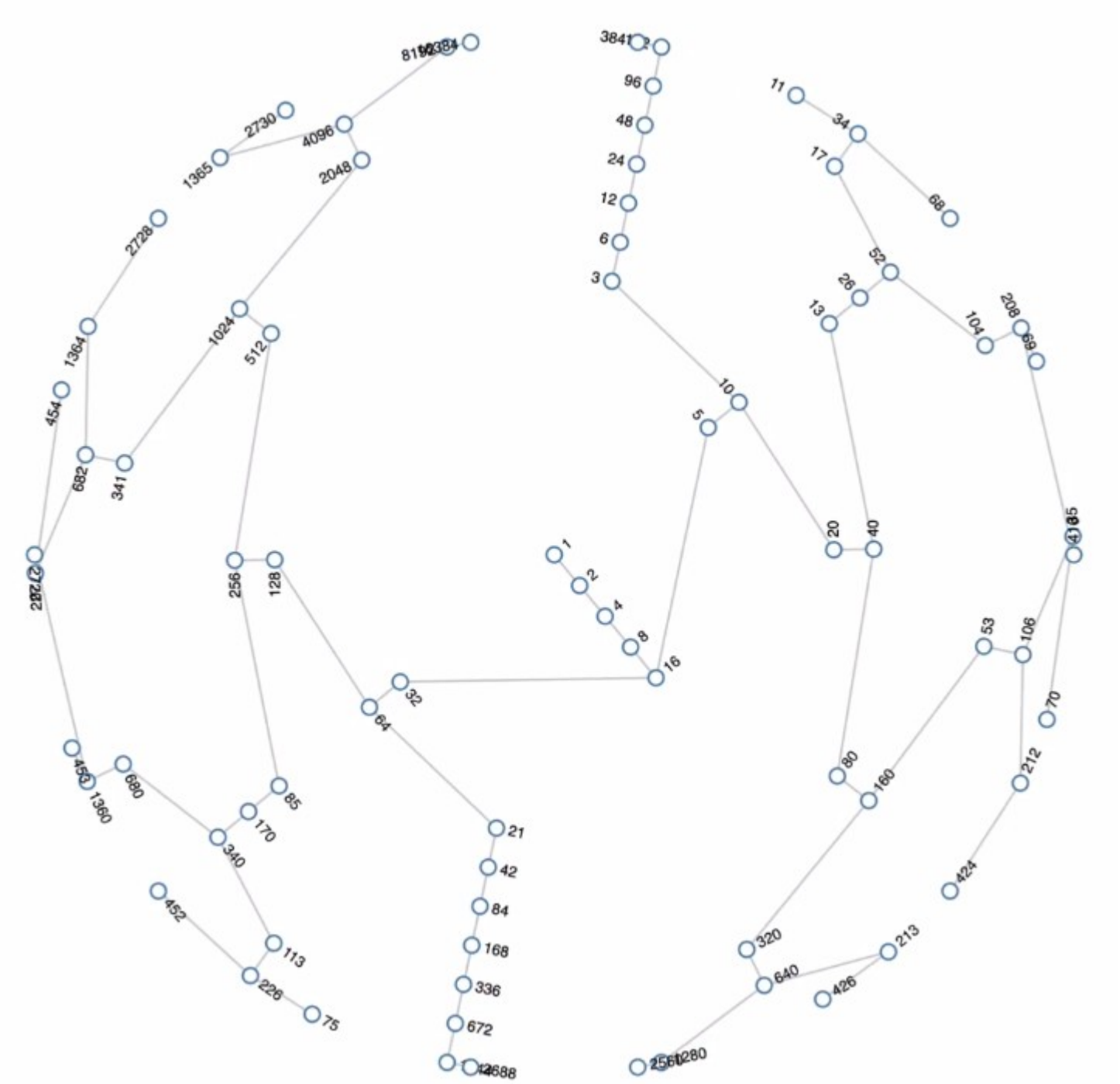
```
def tres_n1(n:int):
    while n>1:
        print (n)
        if n%2==0:
            n = n/2
        else:
            n = 3*n + 1
    print ("Finalizou!")
```

```
tres_n1(9)
```

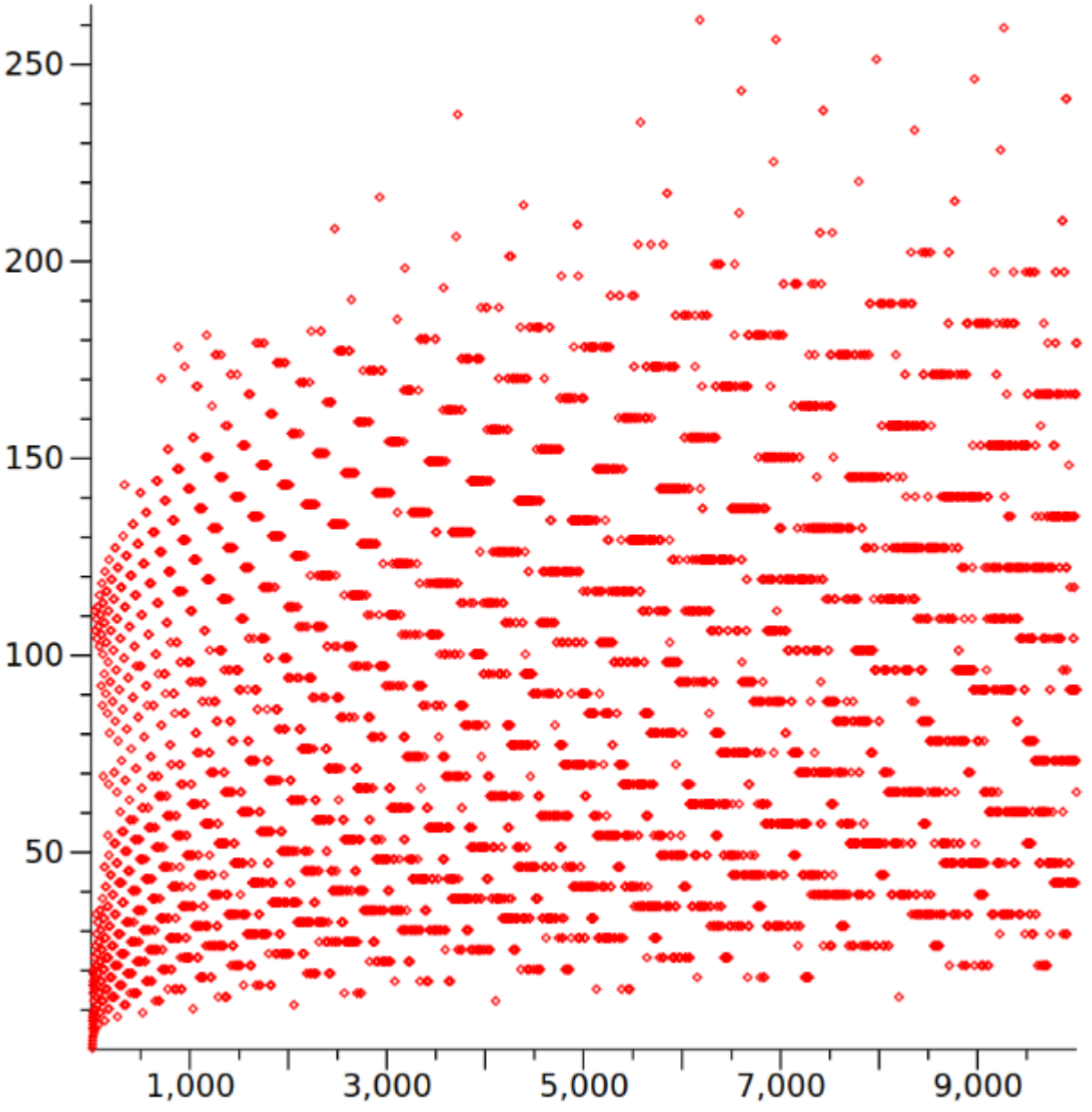
As vezes nem sempre é fácil determinar o número de iterações.

Nesse caso o melhor é usar o laço **while**

Modifique o programa para apresentar o número de passos


8192
4096
2730
1365
2048
2728
1024
512
1364
454
682
341
256
128
1
2
4
8
16
32
64
21
42
84
168
336
672
1344
2688
3841
96
48
24
12
6
3
10
5
20
40
11
34
17
68
52
26
13
104
208
69
85
53
106
70
80
160
212
424
320
640
213
426
1280
2560
453
1360
680
170
85
340
452
113
226
75

Número de passos



Unsolved problem in mathematics:

Does the Collatz sequence eventually reach 1 for all positive integer initial values?

But Tao realized there was something similar about them. With a PDE, you plug in some values, get other values out, and repeat the process — all to understand that future state of the system. For any given PDE, mathematicians want to know if some starting values eventually lead to infinite values as an output or whether an equation always yields finite values, regardless of the values you start with.

For Tao, this goal had the same flavor as investigating whether you always eventually get the same number (1) from the Collatz process no matter what number you feed in. As a result, he recognized that techniques for studying PDEs could apply to the Collatz conjecture.

One particularly useful technique involves a statistical way of studying the long-term behavior of a small number of starting values (like a small number of initial configurations of the water in a pond) and extrapolating from there to the long-term behavior of all possible starting configurations of the pond.

In the context of the Collatz conjecture, imagine starting with a large sample of numbers. Your goal is to study how these numbers behave when you apply the Collatz process. If close to 100% of the numbers in the sample end up either exactly at 1 or very close to 1, you might conclude that almost all numbers behave the same way.

Terence Tao, inspired by a comment on his blog, made some of the biggest progress in decades on the Collatz conjecture.

Courtesy of UCLA

https://www.quantamagazine.org/mathematician-terence-tao-and-the-collatz-conjecture-20191211/

Exercício: Sequência de inteiros

Sequência de inteiros

Crie uma função em que dados dois números inteiros, **n** e **m**, seja impressa uma sequência crescente de ***n x m*** números inteiros.

Para ***n=4***, e ***n=5*** o formato deve ser:

```
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
```

Use apenas UM laço.

Assinatura:

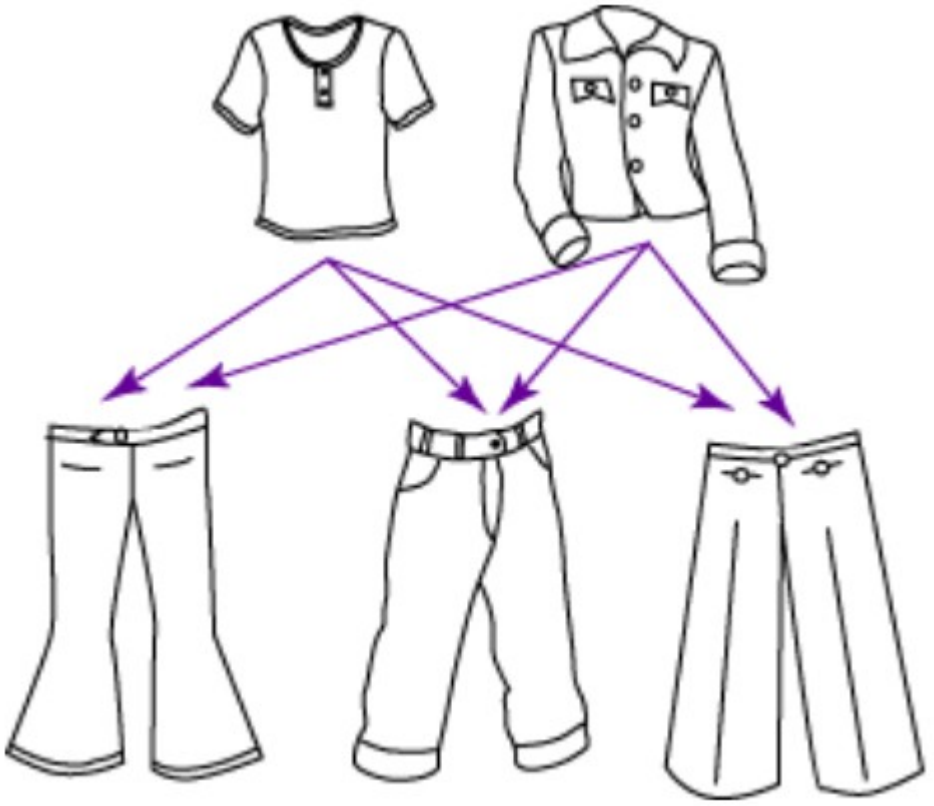
```
def imprimir_sequencia(n:int, m:int):
```

Sequência de inteiros

```
def imprimir_sequencia(n:int, m:int):
    for i in range(1,n*m+1):
        print(i, end=" ")
        if i%m==0:
            print()
```

Trabalhando com laços aninhados

Produto cartesiano:
É a multiplicação entre pares ordenados envolvendo conjuntos distintos.



O que imprime o seguinte trecho?

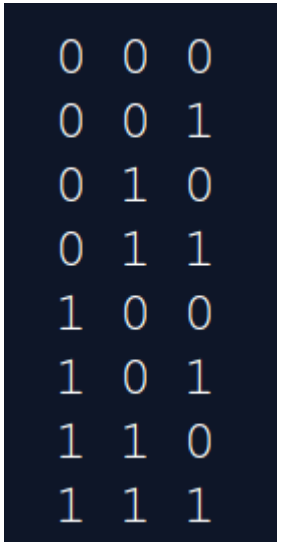
3

```
for i in ['a', 'b', 'c']:
    for j in [1, 2, 3, 4]:
        print(i, j)
```

```
a 1
a 2
a 3
a 4
b 1
b 2
b 3
b 4
c 1
c 2
c 3
c 4
```

O que imprime o seguinte trecho?


```
for i in range(2):
    for j in range(2):
        for k in range(2):
            print(i, j, k)
```


0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

O que imprime o seguinte trecho?


```
n = 5
for i in range(1, n+1):
    for j in range(i, n+1):
        print(i, j)
```

```
1 1
1 2
1 3
1 4
1 5
2 2
2 3
2 4
2 5
3 3
3 4
3 5
4 4
4 5
5 5
```

```
Lista 1:  18/fev   Entrega: 03/mar
Lista 2:  03/mar   Entrega: 09/mar
Lista 3:  10/mar   Entrega: 16/mar
Lista 4:  24/mar   Entrega: 30/mar
Lista 5:  31/mar   Entrega: 06/abr
Lista 6:  07/abr   Entrega: 13/abr
Lista 7:  14/abr   Entrega: 27/mar
Lista 8:  28/abr   Entrega: 12/mai
```

Lista 3:

- **7** exercícios obrigatórios + **1** opcional
- Entrega: 16/março (23h59)

Lista3_1 - Bordas

Crie um programa que permita imprimir apenas as bordas de um retângulo.

Entrada:

O programa recebe dois números inteiros **L**≥1 e **C**≥1 que representam o número de linhas e número de colunas do retângulo.

Saída:

O programa deve desenhar as bordas do retângulo de dimensão **L** x **C**. A parte interna do retângulo deve estar preenchida de espaços em branco.

Exemplos:

Entrada	Saída
3 4	* * * * * * * * * *
7 5	* * * * * * * * * * * * * * * * * * * *
1 1	*

Lista3_2 - Tabuleiro

Crie um programa que permita imprimir uma representação de um tabuleiro quadrado de xadrez.

Entrada:

O programa recebe um número inteiro, maior ou igual a 1, que indica a dimensão do tabuleiro.

Saída:

O programa deve desenhar o tabuleiro com dois caracteres que representam as divisões em cores diferentes conforme exemplos apresentados a seguir

Exemplos:

Entrada	Saída
3	o*o *o* o*o
4	o*o* *o*o o*o* *o*o
5	o*o*o *o*o* o*o*o *o*o* o*o*o