

COLLATZ

1. Importamos las bibliotecas necesarias: `matplotlib.pyplot` y `numpy`.
2. Definimos la función `collatz_sequence(numero)` que toma un número `numero` como entrada y devuelve una lista con la secuencia de Collatz para ese número. La secuencia de Collatz comienza con el número dado y continúa aplicando las siguientes reglas hasta que se llega al número 1:
 - a. Si el número es par, se divide por 2.
 - b. Si el número es impar, se multiplica por 3 y se le suma 1.
3. La función utiliza un bucle `while` para aplicar estas reglas hasta que se llega al número 1. En cada paso de la secuencia, se añade el número actual a la lista secuencia.
4. Pedimos al usuario que ingrese un número y llamamos a la función `collatz_sequence(numero)` para obtener la secuencia de Collatz para ese número.
5. Creamos una lista `x` que va de 0 a `len(secuencia)-1`, es decir, una lista con los índices de la lista secuencia.
6. Creamos una figura y un objeto de ejes `fig, ax` utilizando la función `subplots()` de `matplotlib.pyplot`.
7. Agregamos una línea a la gráfica con los valores de `x` e `y` utilizando la función `plot()` de `matplotlib.pyplot`.
8. Utilizamos un bucle `for` para añadir etiquetas con los valores de la secuencia a la gráfica utilizando la función `annotate()` de `matplotlib.pyplot`.
9. Añadimos etiquetas al eje `x`, al eje `y` y al título de la gráfica utilizando la función `set()` de `matplotlib.pyplot`.
10. Mostramos la gráfica utilizando la función `show()` de `matplotlib.pyplot`.
11. Para que la gráfica se actualice en tiempo real, modificamos la función `collatz_sequence(numero)` para que imprima el número actual de la secuencia y añada un punto rojo a la gráfica utilizando la función `plot()` de `matplotlib.pyplot`. Utilizamos la función `pause()` de `matplotlib.pyplot` para actualizar la gráfica en tiempo real.

```

import matplotlib.pyplot as plt
import numpy as np
import time

def collatz_sequence(numero, ax):
    secuencia = [numero]
    while numero != 1:
        if numero % 2 == 0:
            numero = numero / 2
        else:
            numero = numero * 3 + 1
        secuencia.append(numero)
        print(numero) # Imprime el número actual de la secuencia
        time.sleep(0.5) # Retrasa por 0.5 segundos
        ax.plot(len(secuencia)-1, numero, 'ro') # Agrega un punto rojo a
la gráfica
        plt.pause(0.001) # Actualiza la gráfica en tiempo real
    return secuencia

numero = int(input("Ingrese el número: "))

fig, ax = plt.subplots()
ax.plot([], [])

secuencia = collatz_sequence(numero, ax)

x = np.arange(len(secuencia))
y = np.array(secuencia)

ax.plot(x, y, 'b-')

for i, valor in enumerate(secuencia):
    ax.annotate(str(valor), (x[i], y[i]))

ax.plot(len(secuencia)-1, numero, 'ro') # Agrega un punto rojo a la
gráfica

ax.set(xlabel='Paso', ylabel='Número',
        title='Secuencia de Collatz para {}'.format(numero))
ax.grid()

plt.show()

```



