

Algoritmo de fibonacci iterativo e o gráfico do tempo de execução

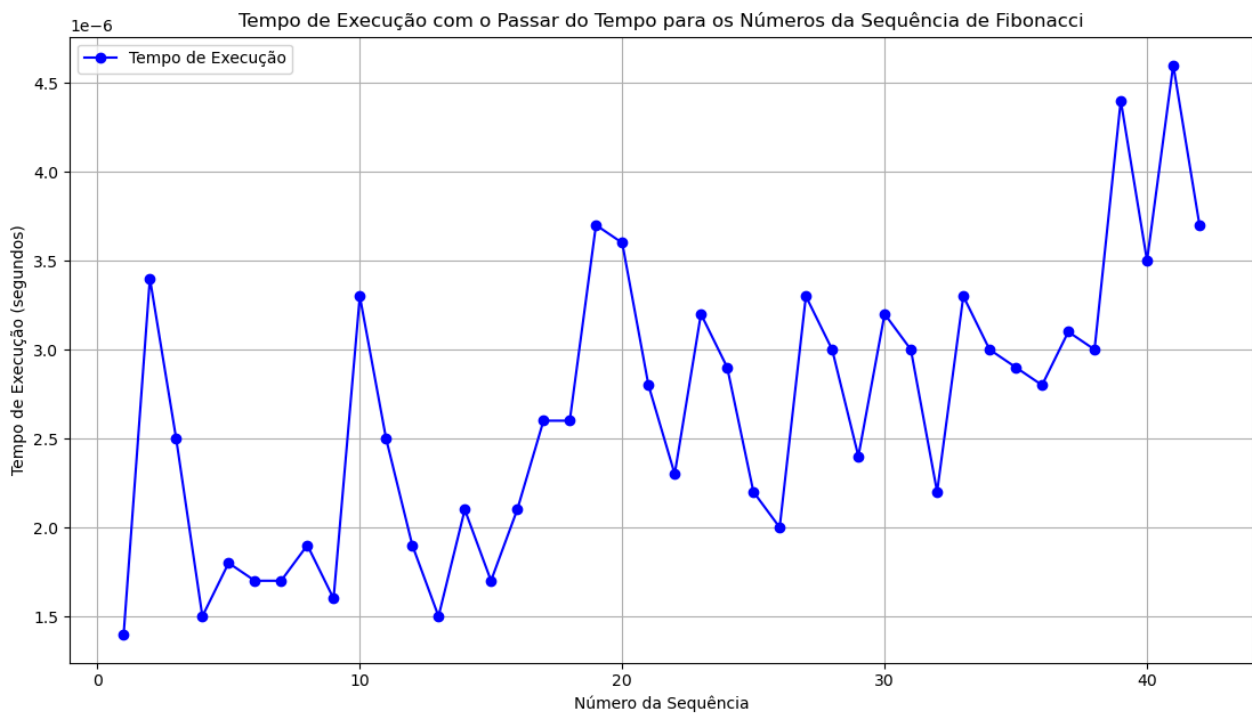
```
import timeit

def fibonacci(n):
    a = 0
    b = 1
    for _ in range(n):
        a, b = b, a + b
    return a

for i in range(1, 43): # 42 é o limite para o cálculo de fibonacci(42) sem causar um tempo de execução maior que 1 minuto

    start_time = timeit.default_timer()
    n_esimo_numero = fibonacci(i)
    end_time = timeit.default_timer()
    execution_time = end_time - start_time

    print(f'{i}º número da sequência de Fibonacci: {n_esimo_numero}')
    print(f'Tempo de execução: {execution_time:.8f} segundos')
```



Algoritmo de Fibonacci recursivo e o gráfico do tempo de execução

```
import timeit

def fibonacci(n):

    if n == 0:
        return 0
    if n == 1:
        return 1

    return fibonacci(n-1) + fibonacci(n-2)

for i in range(1, 43): # 42 é o limite para o cálculo de fibonacci(42) sem causar um tempo de execução maior que 1 minuto

    start_time = timeit.default_timer()
    n_esimo_numero = fibonacci(i)
    end_time = timeit.default_timer()
    execution_time = end_time - start_time

    print(f'{i}º número da sequência de Fibonacci: {n_esimo_numero}')
    print(f'Tempo de execução: {execution_time:.8f} segundos')
```

