

```

1  import timeit
2
3
4  def fibonacci(n):
5      """Non recursive fibonacci function"""
6      for i in range(2, n + 1):
7          fib_list[i] = fib_list[i - 1] + fib_list[i - 2]
8      return fib_list[n]
9
10
11 def fibonacci_recursive(n):
12     """Recursive fibonacci function"""
13     if n == 0:
14         return 0
15     if n == 1:
16         return 1
17     fib_recur_list[n] = fibonacci_recursive(n - 1) + fibonacci_recursive(n - 2)
18     return fib_recur_list[n]
19
20
21 N = 20
22 RUNS = 1000
23 print(f"Given N = {N}\n{RUNS} runs")
24
25 fib_list = [0] * (N + 1)
26 fib_list[0] = 0
27 fib_list[1] = 1
28 print(
29     "Fibonacci non-recursive:",
30     fibonacci(N),
31     "\tTime:",
32     f'{timeit.timeit("fibonacci(N)", setup=f"from __main__ import fibonacci;N={N}",
33         number=RUNS):5f}',
34     "O(n)\tSpace: O(1)",
35 )
36
37 fib_recur_list = [0] * (N + 1)
38 fib_recur_list[0] = 0
39 fib_recur_list[1] = 1
40 print(
41     "Fibonacci recursive:\t",
42     fibonacci_recursive(N),
43     "\tTime:",
44     f'{timeit.timeit("fibonacci_recursive(N)", setup=f"from __main__ import
45     fibonacci_recursive;N={N}", number=RUNS,):5f}',
46     "O(2^n)\tSpace: O(n)",
47 )
48
49 """
50 OUTPUT:
51 Given N = 20
52 1000 runs
53 Fibonacci non-recursive: 6765    Time: 0.001657 O(n)    Space: O(1)
54 Fibonacci recursive:      6765    Time: 2.064246 O(2^n)    Space: O(n)
55 """

```