```python
import timeit


def fibonacci(n):
    """Non recursive fibonacci function"""
    for i in range(2, n + 1):
        fib_list[i] = fib_list[i - 1] + fib_list[i - 2]
    return fib_list[n]


def fibonacci_recursive(n):
    """Recursive fibonacci function"""
    if n == 0:
        return 0
    if n == 1:
        return 1
    fib_recur_list[n] = fibonacci_recursive(n - 1) + fibonacci_recursive(n - 2)
    return fib_recur_list[n]


N = 20
RUNS = 1000
print(f"Given N = {N}\n{RUNS} runs")

fib_list = [0] * (N + 1)
fib_list[0] = 0
fib_list[1] = 1
print(
    "Fibonacci non-recursive:",
    fibonacci(N),
    "\tTime:",
    f'{timeit.timeit("fibonacci(N)", setup=f"from __main__ import fibonacci;N={N}", number=RUNS):5f}',
    "O(n)\tSpace: O(1)",
)

fib_recur_list = [0] * (N + 1)
fib_recur_list[0] = 0
fib_recur_list[1] = 1
print(
    "Fibonacci recursive:\t",
    fibonacci_recursive(N),
    "\tTime:",
    f'{timeit.timeit("fibonacci_recursive(N)", setup=f"from __main__ import fibonacci_recursive;N={N}", number=RUNS,):5f}',
    "O(2^n)\tSpace: O(n)",
)


"""
OUTPUT:

Given N = 20
1000 runs
Fibonacci non-recursive: 6765   Time: 0.001657 O(n)     Space: O(1)
Fibonacci recursive:     6765   Time: 2.064246 O(2^n)   Space: O(n)
"""
```