DAA - Assignment no. 01

Write a program non-recursive and recursive program to calculate Fibonacci numbers and analyze their time and space complexity.

Input:

Assignment1.py

```python
def non_recursive_fibonacci(n):

  if n < 0:
    raise ValueError("n must be a non-negative integer.")
  elif n == 0 or n == 1:
    return n
  else:
    a = 0
    b = 1
    for i in range(2, n + 1):
      c = a + b
      a = b
      b = c
    return c


def recursive_fibonacci(n):

  if n < 0:
    raise ValueError("n must be a non-negative integer.")
  elif n == 0 or n == 1:
    return n
  else:
    return recursive_fibonacci(n - 1) + recursive_fibonacci(n - 2)


def analyze_time_complexity(n):

  non_recursive_time_complexity = "O(n)"

  recursive_time_complexity = "O(2^n)"

  return (f"Non-recursive Fibonacci function time complexity: {non_recursive_time_complexity}\n"
          f"Recursive Fibonacci function time complexity: {recursive_time_complexity}")


```

Assignment1.py ●

Assignment1.py > ...

```
    return (f"Non-recursive Fibonacci function time complexity: {non_recursive_time_complexity}\n"
            f"Recursive Fibonacci function time complexity: {recursive_time_complexity}")


def analyze_space_complexity(n):

    non_recursive_space_complexity = "O(1)"

    recursive_space_complexity = "O(n)"

    return (f"Non-recursive Fibonacci function space complexity: {non_recursive_space_complexity}\n"
            f"Recursive Fibonacci function space complexity: {recursive_space_complexity}")


def main():


    print("Non-recursive Fibonacci numbers:")
    for i in range(10):
        print(non_recursive_fibonacci(i))

    print("\nRecursive Fibonacci numbers:")
    for i in range(10):
        print(recursive_fibonacci(i))

    print("\nTime complexity analysis:")
    print(analyze_time_complexity(10))

    print("\nSpace complexity analysis:")
    print(analyze_space_complexity(10))


if __name__ == "__main__":
    main()

```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Tanmay Mohadikar\Sem 7 Practicals\DAA> & D:/Python/python.exe "d:/Tanmay Mohadikar/Sem 7 Practicals/DAA/Assignment1.py"
Non-recursive Fibonacci numbers:
0
1
1
2
3
5
8
13
21
34

Recursive Fibonacci numbers:
0
1
1
2
3
5
8
13
21
34

Time complexity analysis:
Non-recursive Fibonacci function time complexity: O(n)
Recursive Fibonacci function time complexity: O(2^n)

Space complexity analysis:
Non-recursive Fibonacci function space complexity: O(1)
Recursive Fibonacci function space complexity: O(n)
PS D:\Tanmay Mohadikar\Sem 7 Practicals\DAA>
```