Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
AAD Practical 4

Institute of Computer Technology
B. Tech Computer Science and Engineering

Sub: Algorithm Analysis and Design

Practical 4

Problem : The Fibonacci appears in the smallest, to the largest objects in nature. It is a way for information to flow in a very efficient manner. The number of petals in a flower consistently follows the Fibonacci sequence. Famous examples include the lily, which has three petals, buttercups, which have five, the chicory's 21, the daisy's 34, and so on. Flowers of all kinds follow the pattern, but roses are most favourite kind to use as an example of the Fibonacci Sequence. The petals unfold more & more, and the sequence increases for the best possible exposure to sunlight and other factors. There is a rose flower, which is having only 9 petals. Apply appropriate algorithm/method to find out the sequence for Nth term (9th petal) and also solve them using iteration and recursive method. Compare the performance of two methods by counting the number of steps executed on various inputs. Also draw a comparative chart.

Code :

```
import matplotlib.pyplot as plt
from matplotlib import use
use('GTK3Agg')
import YSL_io as YSL



def rcrsn(n):
if n == 0 or n == 1:
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
AAD Practical 4

```
return n, 1
else:
left, left_count = rcrsn(n - 1)
right, right_count = rcrsn(n - 2)
return left + right, left_count + right_count + 1



def loop(n):
count = 1
a, b = 0, 1

if n == 0 or n == 1:
return n, count

for i in range(n - 1):
a, b = b, a + b
count += 1

return b, count

n = int(YSL.inputGRN("\nEnter nth term to get Fibonacci sequence : "))
YSL.printORNG("\nFibonacci Series: ", end=" ")

count_r = []
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
AAD Practical 4

```
count_l = []
fibonacci_sequence = []

for i in range(1, n + 1):
result, recursion_count = rcrsn(i)
count_r.append(recursion_count)

result, loop_count = loop(i)
count_l.append(loop_count)
fibonacci_sequence.append(result)
print(result, end=", ") if i < n else print(result, end=' ')

data = list(range(1, n + 1))
plt.plot(data, count_l, "-o", label="Iterative Count", color='#d18677',
linewidth=5, markersize=10)
plt.plot(data, count_r, "-o", label="Recursion Count", color='#7b7aaa',
linewidth=5, markersize=10)
YSL.printMGNTA(f'\n\nCounts for {n} terms using loop : ', end='')
print(count_l)
YSL.printBLU(f'\nCounts for {n} terms using recursion : ', end='')
print(count_r)
plt.xlabel("Range of terms", fontsize=15)
plt.ylabel("Number of Iterations", fontsize=15)
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
AAD Practical 4

```
plt.title("Counts of iterations using Recursion and Iterative method to
find Fibonacci Series\n", fontsize=18)
plt.legend()
plt.grid(True)
plt.show()
```

Output :