



Assignment No. 01

Title of the Assignment :

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

Objective of the Assignment :

Students should be able to perform non-recursive & recursive programs to calculate Fibonacci numbers & analyze their time & space complexity.

Prerequisite :

1. Basic of Python or Java Programming.
2. Concept of Recursive and Non-Recursive Functions.
3. Execution flow of calculate Fibonacci numbers.
4. Basic of Time and space complexity.

Contents for Theory :

1. Introduction to fibonacci numbers

2. Time and space complexity

1. Introduction to fibonacci numbers

- In a fibonacci series, every term is the sum of the preceding two terms, starting from 0 & 1 as first & second terms.



What is the Fibonacci Series?

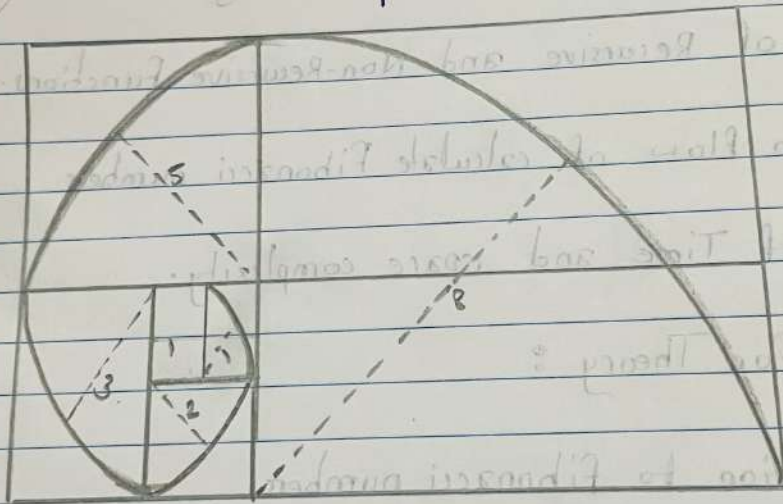
Given the first term, F_0 and second term, F_1 as '0' and '1'.
The third term here can be given as,
 $F_2 = 0 + 1 = 1$

Similarly,

$$F_3 = 1 + 1 = 2$$

$$F_4 = 2 + 1 = 3$$

Given a number n , print n -th Fibonacci Number.



Method 1 (Use Non-recursion)

Program to display the Fibonacci sequence upto n -th term.

`nterms = int(input("How many terms?"))`

first two terms



$n_1, n_2 = 0, 1$

Count = 0

check if the number of terms is valid

if nterms ≤ 0 ;

print("Please enter a positive integer")

if there is only one term, return n_1

elif nterms == 1;

print("Fibonacci sequence upto", nterms, ":")

print(n_1)

generate Fibonacci sequence

else:

print("Fibonacci sequence:")

while count < nterms:

print(n_1)

$n_{th} = n_1 + n_2$

$n_1 = n_2$

$n_2 = n_{th}$

count += 1

Output :

How many terms ? 7

Fibonacci sequence:

0

1

1

2

3

5

8

Time and Space Complexity of Dynamic Programming

- The time complexity of the above code is $T(N)$ i.e., linear. We have to find the sum of two terms & it is repeated n times depending on the value of n .

- The space complexity of the above code is $O(N)$.

- Applications of Fibonacci Series :

- It is used in the grouping of numbers and used to see different patterns found other special mathematical sequences.

- It finds application in coding. For example, Fibonacci series are important in the computational run-time analysis of Euclid's algorithm, used for determining the GCF of two integers.

```
# Program to display
n terms = int(input)
# first two terms
n1, n2 = 0, 1
count = 0
# check if the number
if n terms <= 0:
    print("Please enter a positive integer")
# if there is only one term
elif n terms == 1:
    print("Fibonacci sequence")
    print(n1)
    # generate fibonacci sequence
else:
    print("Fibonacci sequence")
    while count < n terms:
        print(n1)
        nth = n1 + n2
        # update values
        n1 = n2
        n2 = nth
        count += 1
Output
How many terms?
Fibonacci sequence
```

0
1
1
2
3
5
8



- It is applied in numerous fields of science like quantum mechanics, cryptography, etc.

Conclusion :

In this way we have explored Concept of Fibonacci series using recursive and non-recursive method and also learn time & space complexity.

@Adhwan