Experiment. No. 01

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

objective : To perform non-recursive & recursive programs to calculate Fibonacci numbers and analyze their time and space complexity.

Theory :

Introduction to Fibonacci numbers :

The Fibonacci series, named after Italian mathematic-al Lenardo Pisano Bogollo, later known as Fibonacci sequence.

It is denoted by Fn. The numbers in Fibonacci series are given as : 0, 1, 1, 2, 3, 5, 8, 13, 21, 38, ...

In a Fibonacci series, every term is the sum of the precedding two terms, starting from 0 and 1 as first and second first term.

What is Fibonacci series :

A Fibonacci series can thus be given as 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... It can be thus be observed that every term can be calculated by adding the two terms before it.

Given the first term Fo and second term F1 as 0 & 1, the third term can be given as

$F_2 = 0 + 1 = 1$

similarly,

$F_3 = 1 + 1 = 2$

$F_4 = 2 + 1 = 3$

Fibonacci sequence Formula :

The Fibonacci sequence of numbers "Fn" defined using recursive relation with the se values $F_0 = 0$ & $F_1 = 1$:

$$F_n = F_{n-1} + F_{n-2}$$

Here, the sequence is defined using two d part such as kick-off and recursive relation.

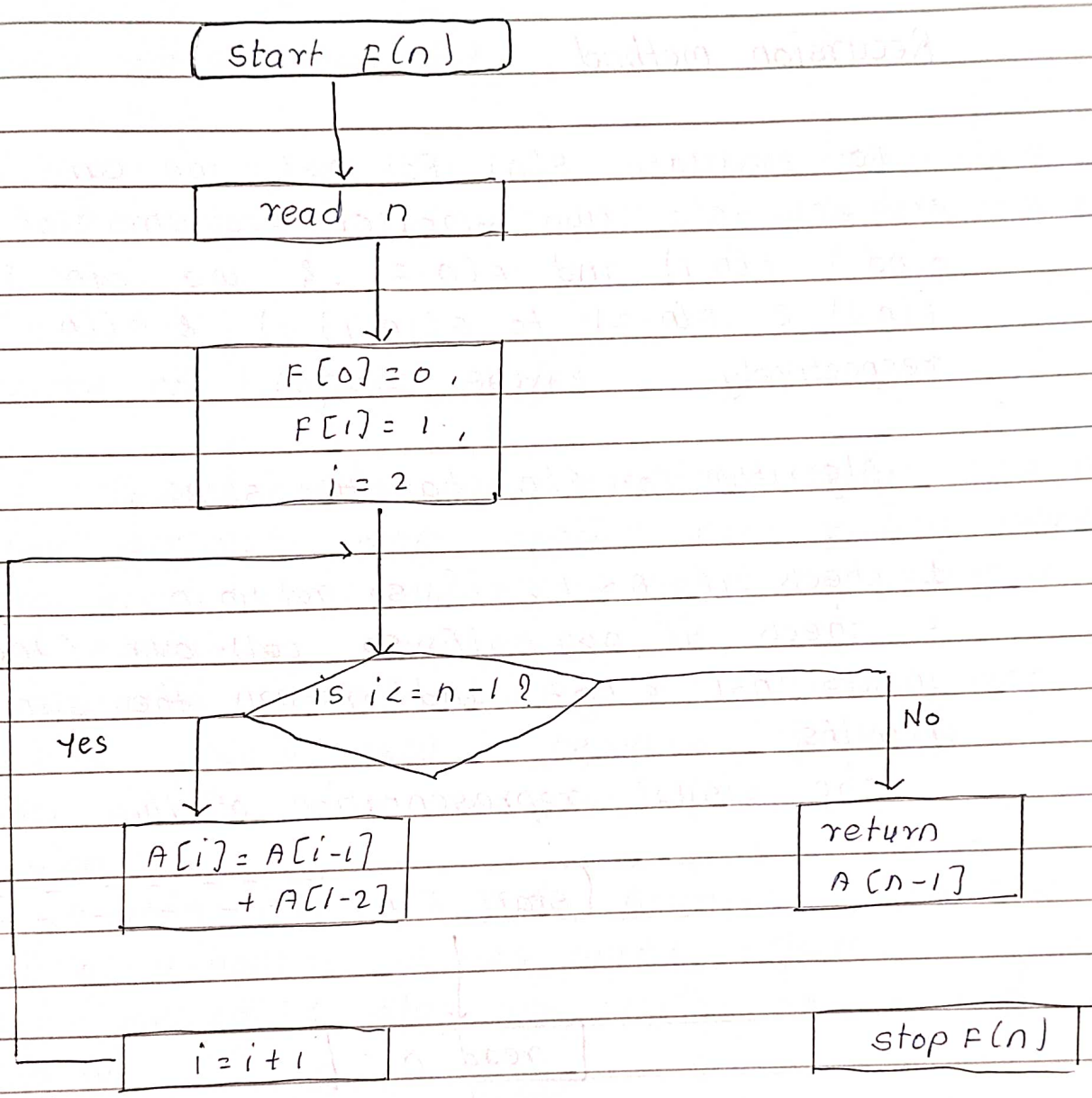
Non-recursion method :

A simple method that is a direct recursive implementation of mathematical recurrence relatio given.

First, we'll store 0 & 1 in F[0] and F[1]. Next, we'll store iterate through array positi 2 to n-1 each position i, we store the s the two preceding array values in F[i].

Finally we return the value of F[n-1], us the number at position n in the sequence.

The visual representation of this process is below



Time and space complexity of space optimized method

The time complexity of Fibonacci series is T(N) i.e. Linear
The space complexity of Fibonacci series is O(1)

Time and space complexity of Dynamic Programming

The time complexity is T(N) i.e. Linear
The space complexity is O(N)

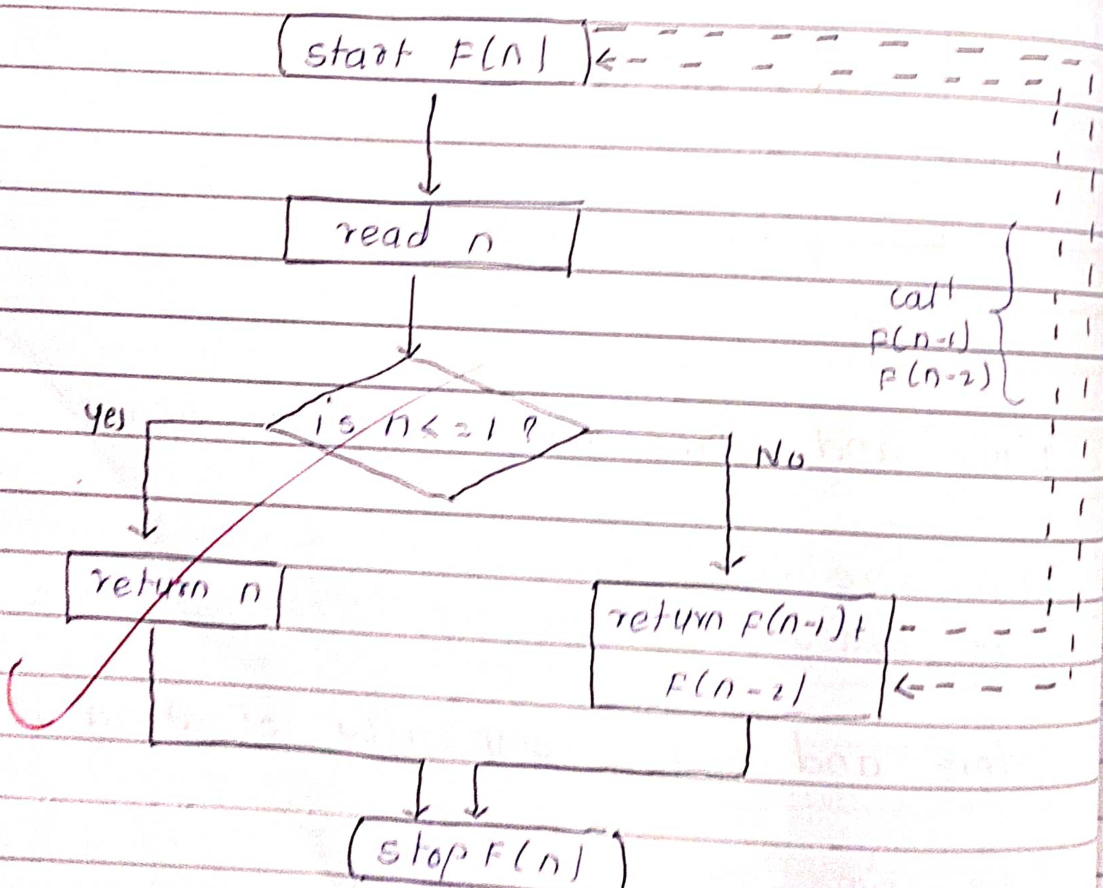
Recursion method

To evaluate F(n) for $n>1$, we can reduce problem into two smaller problems of the sa kind : F(n-1) and F(n-2) , & we can further F(n-1) & F(n-2) to F((n-1)-1) & F((n-1)-2) & F respectively.

Algorithm for F(n) has two steps

1. check if $n \leq 1$, if so return n
2. check if $n>1$, if so call our function F inputs n-1 & n-2, and return the sum of the results

The visual representation of this algorithm



Time and space complexity :

The time complexity is $T(2^N)$ i.e. exponential
The space complexity is $O(N)$ for a recursive series.

Application of Fibonacci series

It is used in the grouping of numbers and used to study different other special mathematical sequences
It is applied to numerous fields of science like quantum mechanics, cryptography, etc.
In Finance market trading, Fibonacci retracement levels are widely used in technical analysis.

Conclusion :

In this way, we have studied & implemented concept of Fibonacci series using recursive & non-recursive method & also find their time & space complexity.