Assignment 2: Recursive  
Function and Efficiency Analysis - Write a recursive function pseudocode and  
calculate the nth Fibonacci number and use Big O notation to analyze its  
efficiency. Compare this with an iterative approach and discuss the pros and  
cons in terms of space and time complexity.

**Pseudocode Iterative:**

function IterativeFibo(n): // Iterative fibonacci

if N = 0:

return 0

else if N = 1:

return 1

else:

a = 0

b = 1

for i = 2 to i = N:

temp = b

b = a + b

a = temp

return b

**Pseudocode Recursive:**

Function RecFibo(n): // RecFibo means recursive fibonacci

If N = 0:

Return 0

Else if N = 1:

Return 1

Else:

Return RecFibo(N-1) + RecFibo(N-2)

**Complexity Calculation :-**

Recursive space complexity : O(n)

Recursive Time complexity : O(2^n)

Iterative space complexity : O(1)

Iterative Time complexity s: O(n)