**Problem Statement**: 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.

**Solution:**

**Method 1: Recursive**

**FUNCTION findFibonacci(int n)**

**IF n <= 1**

**RETURN n**

**ELSE**

**RETURN findFibonocci(n-1) + findFibonocci(n-2)**

**Space Complexity: O(n)**

**Time Complexity: O(2n)**

**Method 2: Iterative**

**FUNCTION findFibonacci(int n) {**

**int =0**

**Int b=1**

**Int fib=0;**

**IF(n<1) {**

**PRINT 1**

**}**

**ELSE {**

**FOR(i=2; i <= n; i++){**

**fib = a+b**

**a = b**

**b = fib }**

**}**

**PRINT fib**

**}**

Time Complexity = O(n)

Space Complexity = O(1)

**Comparison between Recursive and Iterative Method for nth Fibinocci number Psuedo-code:**

|  |  |
| --- | --- |
| **Iterative Method** | **Recursive Method** |
| Number of lines of code are more. | Number of lines of code are less. |
| Time complexity is Linear i.e O(n. | Time complexity is exponential i.e O(2n). |
| Space complexity is constant i.e O(1). | Space complexity is Linear i.e O(n). |

**Pros for Recursive approach:**

* **Code is less complex and easy to understand.**
* **Bigger Problems can be divided into smaller problems.**

**Cons for Recursive approach:**

* **Execution time is more compared to iterative Approach.**
* **It takes more memory as all the recursive call will be stored in stack.**