**Exercise 7: Financial Forecasting**

**Scenario:**

You are developing a financial forecasting tool that predicts future values based on past data.

**Steps:**

1. **Understand Recursive Algorithms:**
   * Explain the concept of recursion and how it can simplify certain problems.

* **Concept of Recursion:**

Recursion is a method where a function calls itself directly or indirectly to solve a problem. It simplifies certain problems by breaking them down into smaller, more manageable sub-problems. Each recursive call should bring the problem closer to a base case, which is solved directly without further recursion.

* **Example:**

A classic example of recursion is the calculation of the factorial of a number n:

n!=n×(n−1)!

This can be implemented recursively as:

public int factorial(int n) {

if (n <= 1) { // Base case

return 1;

} else { // Recursive case

return n \* factorial(n - 1);

}

}

1. **Setup:**
   * Create a method to calculate the future value using a recursive approach.
2. **Implementation:**
   * Implement a recursive algorithm to predict future values based on past growth rates.
3. **Analysis:**
   * Discuss the time complexity of your recursive algorithm.

**Time Complexity:**

* The time complexity of this recursive algorithm is O(n), where n is the number of periods. This is because each recursive call decreases the number of periods by 1, leading to a linear number of calls.
  + Explain how to optimize the recursive solution to avoid excessive computation.

Recursion can lead to excessive computation and stack overflow for large inputs. To optimize, we can use memoization or convert the recursion to an iterative approach. Here’s how to convert the above recursive solution to an iterative one:

public class FinancialForecasting {

// Method to calculate future value iteratively

public double calculateFutureValueIterative(double initialValue, double growthRate, int periods) {

double futureValue = initialValue;

for (int i = 0; i < periods; i++) {

futureValue \*= (1 + growthRate);

}

return futureValue;

}

public static void main(String[] args) {

FinancialForecasting forecasting = new FinancialForecasting();

double initialValue = 1000.0; // Initial value

double growthRate = 0.05; // Growth rate (5%)

int periods = 10; // Number of periods

double futureValue = forecasting.calculateFutureValueIterative(initialValue, growthRate, periods);

System.out.println("Future Value: " + futureValue);

}

}