**Exercise 7: Financial Forecasting**

**1. Understand Recursive Algorithms**

**Recursion:**

**Definition: Recursion is a process in which a function calls itself directly or indirectly to solve a problem. Each recursive call solves a smaller instance of the problem.**

**Base Case and Recursive Case:**

**Base Case: The condition under which the recursion ends.**

**Recursive Case: The condition where the function continues to call itself.**

**Simplifies code for problems that have a naturally recursive structure, such as tree traversals, factorial calculation, and Fibonacci sequence. Provides a clear and concise way to solve problems that can be broken down into similar subproblems.**

**3. Implementation**

**public class FinancialForecasting {**

**// Recursive method to calculate future value**

**public static double calculateFutureValue(double currentValue, double growthRate, int years) {**

**// Base case: if no years left, return the current value**

**if (years == 0) {**

**return currentValue;**

**}**

**// Recursive case: calculate future value for remaining years**

**return calculateFutureValue(currentValue \* (1 + growthRate), growthRate, years - 1);**

**}**

**public static void main(String[] args) {**

**double currentValue = 1000; // initial value**

**double growthRate = 0.05; // 5% annual growth rate**

**int years = 10; // number of years**

**double futureValue = calculateFutureValue(currentValue, growthRate, years);**

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

**}**

**}**

**4. Analysis**

**Time Complexity:**

**The time complexity of the recursive algorithm is O(n), where n is the number of years. Each recursive call processes one year, leading to n calls.**

**Optimization:**

**Memoization: Store results of previous calculations to avoid redundant computations. This can be done using a map or array.**

**Dynamic Programming: Use an iterative approach to fill in a table of values from the base case up to the desired value, avoiding recursion altogether.**