**Exercise 7: Financial Forecasting**

1. **Understand Recursive Algorithms:**

* Explain the concept of recursion and how it can simplify certain problems.

Ans. Recursion is a technique used in computer science to solve big problems by breaking them into smaller, similar problems. It involves calling a function repeatedly until a base condition is met. Recursion can simplify complex problems by breaking them down into smaller, more manageable pieces, and is essential for some algorithms and data structures

**2.Setup:**

* + Create a method to calculate the future value using a recursive approach.

**3.Implementation:**

* + Implement a recursive algorithm to predict future values based on past growth rates.

Code :

package Algorithm\_DataStructures.Exercise7.Financial\_Forecasting;

public class FutureValueCalculator {

    // Recursive method to calculate the future value based on past growth rates

    public static double calculateFutureValue(double presentValue, double growthRate, int years) {

        if (years == 0) {

            return presentValue;

        } else {

            double futureValue = presentValue \* (1 + growthRate);

            return calculateFutureValue(futureValue, growthRate, years - 1);

        }

    }

    public static void main(String[] args) {

        double presentValue = 1000.0; // Initial investment

        double growthRate = 0.05; // Growth rate per year

        int years = 5; // Number of years into the future

        double futureValue = calculateFutureValue(presentValue, growthRate, years);

        System.out.printf("The future value after %d years is: %.2f", years, futureValue);

    }

}

1. **Analysis:**

* *Discuss the time complexity of your recursive algorithm.*

Ans. In the provided recursive algorithm for calculating future values based on past growth rates, the time complexity can be analyzed as follows:

* The recursive method calculateFutureValue is called years times recursively, each time performing a constant number of operations.
* Therefore, the time complexity of this recursive algorithm is O(years), where years is the number of years into the future for which the future value is being calculated.
* *Explain how to optimize the recursive solution to avoid excessive computation.*

Ans. While the recursive approach is straightforward and intuitive, it can lead to excessive computation for larger values of years. To optimize the recursive solution and avoid unnecessary computation, you can implement memorization. Memorization is a technique that stores the results of expensive function calls and returns the cached result when the same inputs occur again.