**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.
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.
4. **Analysis:**
   * Discuss the time complexity of your recursive algorithm.
   * Explain how to optimize the recursive solution to avoid excessive computation.

Solutions:

1. **Understand Recursive Algorithms:**

Recursion is where a method invokes itself to find a solution to a smaller instance of the same problem until it reaches a base case. It makes it easy to handle problems such as factorial, Fibonacci, and in this example, computing future values annually. Rather than applying loops, recursion divides the work into smaller repeated operations, easier to comprehend and code.

**Setup and Implementation (2 & 3):**

simple Java code to predict future value based on an annual growth rate using recursion:

import java.util.Scanner;

public class FinancialForecast {

    //recursive method to calculate future value

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

        if (years == 0) {

            return currentValue;

        }

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

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        //input from user

        System.out.print("Enter initial amount: ");

        double initialValue = scanner.nextDouble();

        System.out.print("Enter annual growth rate (e.g., 0.05 for 5%): ");

        double growthRate = scanner.nextDouble();

        System.out.print("Enter number of years: ");

        int years = scanner.nextInt();

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

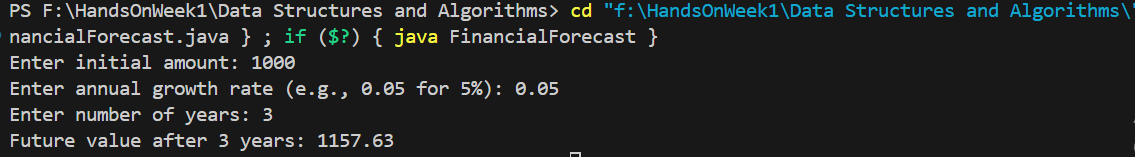
        System.out.printf("Future value after %d years: %.2f\n", years, futureValue);

        scanner.close();

    }

}

Output:



4.

The recursive method utilized in the financial forecasting tool is already optimized because it only makes one recursive call in each step with no branching or multiple calls. This implies that it does not repeat redundant calculations of old values or redundant steps. However, because every recursive call utilizes stack space, it can be optimized further by rewriting it as an iterative method through a basic loop. The iterative one employs a for loop to multiply the amount every year by the growth rate, achieving the same result without any recursion and requiring extra memory. This is more memory frugal and secure for large numbers of years.

public static double futureValueIterative(double amount, double rate, int years) {

    for (int i = 0; i < years; i++) {

        amount \*= (1 + rate);

    }

    return amount;

}