**Exercise 7: Financial Forecasting**

**1. Understanding Recursive Algorithms**

Recursion is a powerful programming technique where a method calls itself to solve smaller instances of the same problem. For financial forecasting, recursion provides an elegant way to model compound growth by breaking down the calculation into smaller, identical steps.

**Key Characteristics:**

* **Base Case:** The simplest scenario with a direct solution (e.g., 0 periods remaining)
* **Recursive Case:** The method calls itself with modified parameters (e.g., reduced time periods)
* **Termination:** Each recursive call must progress toward the base case.

**2,3. Setup and Implementation**

public class FinancialForecast {

// Recursive method to calculate future value

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

if (years == 0) {

return presentValue; // Base case: no more years left

}

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

}

// Main method to test the forecasting

public static void main(String[] args) {

double initialInvestment = 10000.0; // Example: $10,000

double annualGrowthRate = 0.05; // Example: 5% growth per year

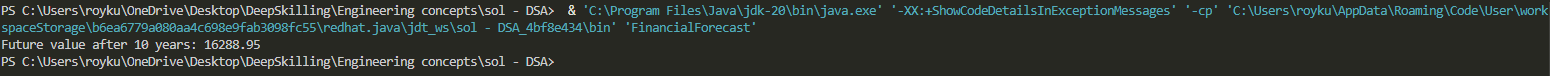
int numberOfYears = 10; // Forecast for 10 years

double futureValue = calculateFutureValue(initialInvestment, annualGrowthRate, numberOfYears);

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

}

}

**Output**

**4. Analysis**

**Time Complexity:**

* **Basic Recursive**: O(n) time and space (due to call stack)
* **Forecast with History:** O(n²) for simple average (can be optimized to O(n))
* **Optimized Iterative:** O(n) time, O(1) space

**Memory Considerations:**

* Each recursive call adds a stack frame
* Deep recursion may cause StackOverflowError
* Iterative approach avoids stack issues

**Optimization Strategies:**

1. **Memorisation:**

* Cache intermediate results to avoid redundant calculations
* Particularly useful for multiple forecasts with same parameters

1. **Iterative Conversion:**

* Replaces recursion with loops to prevent stack overflow
* More memory efficient for large periods

1. **Precision Handling:**

* Uses BigDecimal for accurate financial calculations
* Avoids floating-point rounding errors

1. **Growth Rate Calculation:**

* Multiple methods (simple average, exponential, recent trend)
* Each suitable for different market conditions