**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.

**Solution :**

 Analysis

1. **Time Complexity**:
   * **Recursive Method**: Each recursive call reduces periods by 1 until it reaches 0. For n periods, there are n recursive calls.
   * **Complexity**: **O(n)** (linear time), where n is the number of future periods.
2. **Optimization**:
   * **Problem**: The naive recursion has **O(n)** space complexity due to recursion stack depth, risking StackOverflowError for large n.

**Solutions**:

* **Iterative Approach**: Replace recursion with a loop to avoid stack overhead:  
    
  private static double predictFutureValueIterative(double currentValue, double growthRate, int periods) {

double result = currentValue;

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

result \*= (1 + growthRate);

}

return result;

}

* **Closed-Form Formula**: Directly compute using exponentiation for **O(1)** time:  
    
  private static double predictFutureValueFormula(double currentValue, double growthRate, int periods) {

return currentValue \* Math.pow(1 + growthRate, periods);

}

.