**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.

* Recursion is when a function calls itself to solve smaller parts of a problem

1. **Setup:**

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

Let’s assume we have:

* An initial value (like revenue)
* A fixed annual **growth rate**
* Predicting the value after **N years**

**Formula (Compound Growth):**

FutureValue(n)​=FutureValue(n−1)​×(1+growthRate)

1. **Implementation:**

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

public class FinancialForecaster {

public double predictFutureValue(double initialValue, double growthRate, int years) {

if (years == 0) {

return initialValue;

}

return predictFutureValue(initialValue, growthRate, years - 1) \* (1 + growthRate);

}

public static void main(String[] args) {

FinancialForecaster forecaster = new FinancialForecaster();

double initialValue = 1000.0;

double growthRate = 0.10;

int years = 5;

double futureValue = forecaster.predictFutureValue(initialValue, growthRate, years);

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

}

}

1. **Analysis:**

Discuss the time complexity of your recursive algorithm.

Time complexity = O(n)

Explain how to optimize the recursive solution to avoid excessive computation.

Converting to iterative approach:

public double predictFutureValueIterative(double initialValue, double growthRate, int years) {

double result = initialValue;

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

result \*= (1 + growthRate);

}

return result;

}