**QUESTION:**

**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.

**UNDERSTANDING:**

Recursion is a function that repeatedly calls itself. It simplifies problem by breaking them down into further smaller problems for faster and more manageable outputs.

For example, problems like Tower of Hanoi, tree traversals, Factorial and Fibonacci sequence can be easily solved using recursion.

**CODE:**

 public class Financialforecasting {

    public static void main(String[] args) {

        int[] pastData = {100, 120, 130, 150, 170, 200};

        double growthRate = 10.2;

        int years = 5;

        System.out.println("Setting Sample growth rate to "+growthRate+"% and years to "+years+"...");

        System.out.println("Calculating future value based on past data...");

        System.out.println("Output using recursive function: "+calculateFutureValue(pastData[pastData.length-1], growthRate/100, years));

    }

    public static double calculateFutureValue(int pastvalue, double growthrate, int years) {

        if (years == 0)

            return pastvalue;

        else

            return calculateFutureValue(pastvalue, growthrate, years - 1)\*(growthrate+1);

    }

}

**OUTPUT:**

A screenshot of a computer

AI-generated content may be incorrect.

**CONCLUSION:**

For each n input (years) the algorithm recursively calls itself n times. So the time complexity is **O(n)**.

We can optimize the recursive function by using an Iterative Approach. Although it still has the same time complexity **O(n)** it is comparatively faster compared to recursive approach. Also iterative approach is the most efficient for large data inputs.