**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.

Ans: Recursion is a programming technique where a function calls itself directly or indirectly in order to solve a problem. Recursion can simplify certain problems by breaking them down into smaller, more manageable sub-problems that resemble the original problem. For example, calculating the factorial of a number or solving the Fibonacci sequence can be elegantly handled with recursion.

***Setup:***

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

Code: Growth.java

***Implementation:***

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

Code: Growth.java

***Analysis:***

Discuss the time complexity of your recursive algorithm.

**Ans:***Time Complexity:* The time complexity of the recursive algorithm for calculating future value is O(n)O(n)O(n), where nnn is the number of periods. This is because the function makes one recursive call for each period until it reaches the base case.

*Optimizing the Recursive Solution:* To avoid excessive computation and potential stack overflow issues with deep recursion, we can use **memoization** or an **iterative approach**.

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

***Ans:*** We can convert the recursive method to an iterative one to avoid the overhead of recursive calls and improve performance.