**Exercise 7: Financial Forecasting**

**1. Understanding Recursive Algorithms**

**Concept of Recursion:**

Recursion is a technique where a function calls itself in order to solve smaller instances of the same problem. It simplifies problems by breaking them down into more manageable sub-problems. Recursion is especially useful for problems that exhibit a repetitive structure or can be divided into similar sub-problems.

**How Recursion Simplifies Problems:**

**Divide and Conquer:** Recursion divides the problem into smaller sub-problems, which are easier to solve. Each recursive call works on a smaller subset of the original problem.

**3. Implementation**

In the calculateFutureValue method, recursion is used to compute the future value of an investment over several periods. Each call to calculateFutureValue reduces the number of periods by one until it reaches the base case (when periods is 0). At each step, it multiplies the present value by (1 + growthRate) to get the value for the next period, demonstrating how recursion simplifies the calculation of compound growth.

**4. Analysis of the Recursive Algorithm**

**Time Complexity:**

The time complexity of the recursive algorithm, as provided, is O(n), where n is the number of periods. This is because, in the worst case, the function will make one recursive call for each period. However, due to memoization, each unique number of periods is calculated only once, and subsequent calls for the same number of periods retrieve the result from the memoization cache.

**Optimizing Recursive Solutions:**

Memoization is used in the provided code to avoid redundant calculations, which significantly optimizes the recursive approach. By storing the results of previously computed values in a HashMap, the algorithm avoids recalculating the future value for the same number of periods multiple times. This reduces the number of recursive calls and prevents excessive computation.