**Financial Forecasting**

**1. Understanding Recursive Algorithms**

**What is Recursion?**

Recursion means a function that calls is on itself in order to sort out the smaller portions of a problem. It helps by increasing the problem’s tractability — its ability to be decomposed into a set of smaller problems. This method is quite convenient when the problem at hand has some resemblance to some other problem or can be broken down into similar problems.

**How Recursion Simplifies Problems**

Divide and Conquer: Recursion divide a problem into several solvable smaller problems by calling upon itself in the solution process. Each time the function is called it works on a part of the original problem, thus moving from one iteration to another until a solution is arrived at.

**Implementation**

Recursion is used in the calculateFutureValue method to solve the future value of an investment for multiplicative periods. Yes, the function uses recursion to call the function and with each call, the number of periods is decremented by 1 until the base case is arrived when there is 0 period. At each step, to arrive at the value for the next period, it multiplies the present value by (1+ growthRate). This process demonstrates how recursion can be used to make calculations of compound growth easy.

**The identification of the Recursive Algorithm**

**Time Complexity**

Since, this is a recursive algorithm, therefore, the time complexity in the worst case is O(n), where n is the number of periods. This is because, at the worst, the function makes call to itself only once per period in the worse case. However, because of memoization the task of calculating each unique period length is done only once. This means that after the first calculation, the outcomes are saved for reuse reducing the number of calculations done.

**Optimizing Recursive Solutions**

The use of memoization is defined in the code as one of the optimization techniques used commonly. Using the HashMap, the algorithm stores the results of previously computed values to avoid calculating the future value for the said number of periods more than once. This alters the way the recursive calls are made and helps to minimize the number of such calls which in turn eliminates unnecessary computations, thus improving the efficiency of the end solution.