**Exercise 7: Financial Forecasting**

**Explain the concept of recursion and how it can simplify certain problems.**

Recursion is a process where a function calls itself directly or indirectly to solve a problem. Base Case is the condition under which the recursion stops. Recursive Case is the condition where the function calls itself with modified arguments.

Recursion can simplify certain problems by breaking them down into smaller, more manageable subproblems that have the same structure as the original problem. This divide-and-conquer approach allows for a more straightforward and often more intuitive solution. Each recursive call solves a smaller instance of the problem until it reaches a base case. This makes the solution more manageable and easier to understand.

**Discuss the time complexity of your recursive algorithm.**

The time complexity of the recursive algorithm is O(n), where n is the number of periods. Each recursive call reduces the number of periods by one until it reaches the base case.

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

**Memoization:** It is a technique used to store the results of expensive function calls and reuse them when the same inputs occur again. This helps in reducing the number of computations and improving the efficiency of the algorithm. Store the results of subproblems to avoid redundant calculations.

**Iterative Approach:** It converts the recursive algorithm into an iterative one, often using loops. This avoids the overhead of recursive calls and the potential for stack overflow. Iterative solutions are usually more memory-efficient because they don't involve the call stack. Convert the recursive solution to an iterative one to avoid stack overflow and reduce memory usage.