**Analysis of Recursive Algorithm - Financial Forecasting**

**Understanding Recursive Algorithms:**

Recursion is a programming technique where a function calls itself to solve smaller subproblems of a larger problem. It simplifies complex problems by breaking them down into base cases and recursive steps. For example, in financial forecasting, future values can be calculated recursively by applying growth repeatedly over time.

**Time Complexity:**

The recursive function for forecasting has a time complexity of O(n), where 'n' is the number of years. Each recursive call processes one year, resulting in a linear number of calls. This is efficient for small input sizes, but may become slower for large values of 'n' due to repeated function calls and stack usage.

**Optimization Techniques:**

To avoid excessive computation and stack overflow in recursion, the following optimizations can be used:  
1. Convert the recursion to an iterative loop, which avoids deep call stacks and improves performance.  
2. Use memoization (caching) to store and reuse previously computed results in problems involving repeated subproblems.  
3. Use tail-recursion (if supported by the language), which allows the compiler to optimize recursion internally.