# Exercise 7: Financial Forecasting

## Understanding Recursive Algorithms

Recursion is a programming technique where a function calls itself to solve smaller instances of a problem. This approach is especially useful for problems that exhibit repetitive patterns or can be broken down into sub-problems. Recursion simplifies the code and helps manage complex logic with minimal lines.

## **Analysis**

Time Complexity:

In its basic form, the recursive approach has a time complexity of O(n), where n is the number of years. This is because each call reduces the problem by one year until it reaches the base case.

**Optimization:**

To prevent excessive computation and stack overflow issues in deep recursion, the method can be optimized using:  
• Memoization – store results of already computed years to avoid redundant calculations.  
• Iterative approach – convert recursion to a loop if recursion depth is a concern.  
• Tail Recursion – a form of recursion that compilers can optimize, reducing call stack usage.

## Conclusion

Recursive algorithms offer a clean and logical way to solve forecasting problems. When optimized properly, they are effective for predicting future financial values based on historical trends. In production systems, a combination of recursion with optimization techniques ensures both accuracy and performance.