**Exercise 7: Financial Forecasting**

**1. Understand Recursive Algorithms**

**Recursion** is a programming technique where a function calls itself in order to solve a problem. It is especially useful for problems that can be broken down into smaller, similar subproblems. In recursion, a base case is defined to stop the recursion, and each recursive call should reduce the problem toward that base case.

**How recursion simplifies problems:**

* Reduces complex problems to simpler subproblems.
* Makes code shorter.

In the context of financial forecasting, recursion can be used to repeatedly apply a growth formula year-over-year, rather than using loops.

**2. Analysis**

**Time Complexity:**

* The recursive solution has a time complexity of **O(n)**, where *n* is the number of periods (e.g., years).
* Each recursive call performs one multiplication and proceeds to the next year, resulting in *n* calls in total.

**Optimization Considerations:**

* For large values of *n*, recursion can lead to stack overflow due to deep recursion.
* To optimize:
  + Convert the recursion to an **iterative loop**, reducing space complexity to **O(1)**.
  + We can use the concept of dynamic programming in many cases.

**Conclusion:**  
Recursive algorithms provide a clear and intuitive way to implement financial forecasting logic. However, for efficiency and scalability, converting to an iterative approach is recommended for production environments dealing with large time spans.