**Exercise 7: Financial Forecasting (Theory Answers)**

**Step 1: Understanding Recursive Algorithms**

**What is recursion, and how can it simplify certain problems?**

Recursion is a programming technique where a function calls itself to solve a smaller version of the original problem. It is especially useful when a problem has a naturally repetitive or self-similar structure.

In financial forecasting, recursion can simplify the process by:

* Eliminating the need for loops in repeated calculations.
* Providing a clear and logical structure for modeling year-by-year financial growth.
* Allowing developers to focus more on logic and less on control flow.

For example, when predicting future values year by year based on a growth rate, recursion naturally reflects this progressive structure.

**Step 4: Analysis**

**What is the time complexity of the recursive algorithm?**

The time complexity is **O(n)**, where *n* is the number of years. Each recursive call processes one year and reduces the problem by 1 until it reaches the base case.

**How can the recursive solution be optimized to avoid excessive computation?**

* **Convert to Iterative Approach**: Use a loop instead of recursion to avoid potential stack overflow for large n.
* **Tail Recursion**: If the language/compiler supports it, tail recursion can reduce stack usage.
* **Memoization**: Useful in recursive problems with overlapping subproblems (not applicable here, since each year is unique).

**Conclusion**: While recursion is simple and elegant for small inputs, iterative or formula-based approaches are more efficient and safer for large time periods.