**Exercise 7: Financial Forecasting**

**Scenario:**

You are developing a financial forecasting tool that predicts future values based on past data.

* Recursion is a programming technique where a function calls itself to solve smaller instances of the same problem. This process continues until it reaches a base case, which stops the recursion.
* Recursion works best when a problem is self-similar—meaning that the entire problem looks like a smaller version of itself. In such problems, solving the big problem depends on solving smaller instances of the same problem.
* Recursion allows you to divide a complex problem into smaller, more manageable pieces. Instead of solving everything at once, you solve a smaller version of the problem, and repeat the process until it becomes simple enough to solve directly (base case).



**Time and Space Complexity**

**The time complexity of the recursive algorithm for calculating future value is O(n)**, where n is the number of years. This is because the function calls itself once for each year, performing a constant amount of work during each call (a multiplication and a subtraction). Therefore, the total number of recursive calls grows linearly with the number of years. **The space complexity is also O(n)** due to the call stack storing information for each recursive call until the base case is reached.

* In more complex recursive problems where subproblems overlap or calculations are repeated, memoization can be applied to store and reuse previously computed results.
* This reduces redundant computations, improving efficiency. However, for simple linear recursive problems like the future value calculation, memoization is not necessary and the iterative approach is typically preferred due to its simplicity and safety.