**Financial Forecasting**

**Understanding Recursive Algorithms**

**Recursion** is a programming technique where a function calls itself directly or indirectly to solve a problem. Recursion simplifies problems by breaking them down into smaller, more manageable sub-problems of the same type.

In financial forecasting, recursion can be used to predict future values based on past data by repeatedly applying a growth rate to a known value.

**Setup**

We need a method to calculate the future value using a recursive approach. For simplicity, let's assume the future value is based on a constant growth rate.

Here's the formula for future value with a growth rate: Future Value=Present Value×(1+Growth Rate)n\text{Future Value} = \text{Present Value} \times (1 + \text{Growth Rate})^nFuture Value=Present Value×(1+Growth Rate)n where nnn is the number of periods.

**Analysis**

**Time Complexity:**

The time complexity of the recursive algorithm is O(n)O(n)O(n), where nnn is the number of periods. This is because the function makes one recursive call for each period until it reaches the base case.

**Optimizing the Recursive Solution:**

Recursion can lead to excessive computation if not optimized. In this case, our recursive method does not have overlapping subproblems, so memoization isn't necessary. However, to improve efficiency and avoid deep recursion (which could lead to a stack overflow for large nnn), we can convert the recursive method to an iterative one.