**Understanding Recursive Algorithms in Forecasting**

**Recursion** is a programming technique where a method calls itself to solve smaller parts of a problem. This continues until it reaches a base case, making it easier to solve problems that follow a repeated pattern or depend on previous steps.

In forecasting scenarios (like predicting investment growth), recursion helps represent growth that compounds over time. If a value grows by a constant rate r each year, we can define the future value in recursive form:

**FutureValue(n) = (1 + r) × FutureValue(n - 1)**

This breaks down the problem year by year, making the logic easier to implement and understand for scenarios like financial growth, interest, or population prediction.

**Time Complexity**

The recursive function performs one multiplication per call and reduces the year count by 1 in each step. So for n years, it will make n calls.

**Time Complexity = O(n)**  
This means the time grows linearly with the number of years.

**Optimization**

In general, recursion can be slow when the same subproblems repeat. In such cases, we use **memoization** to store and reuse results.

But in forecasting, each year gives a unique result, so no subproblem is repeated. However, making too many recursive calls can lead to a **stack overflow** if n is very large.

To avoid this, we can switch to an **iterative solution**, which calculates the same result using loops. This improves performance and prevents memory issues.