**Financial Forecasting**

**Understand Recursive Algorithms**

Concept of Recursion

* Definition: Recursion is a method where a function calls itself in order to solve smaller instances of the same problem. The key components of a recursive algorithm are:
  + Base Case: The condition under which the recursion stops.
  + Recursive Case: The part where the function calls itself with modified arguments.
* Simplification: Recursion can simplify complex problems by breaking them down into smaller, more manageable problems. It is especially useful for problems that exhibit overlapping subproblems and optimal substructure, such as those found in dynamic programming.

**Time Complexity**

* Time Complexity: The recursive function calculateFutureValue makes a recursive call for each year. Hence, the time complexity is O(n), where n is the number of years. Each recursive call performs a constant amount of work (multiplying and adding), so the time complexity is linear with respect to the number of years.

**Optimization**

Recursive algorithms can lead to excessive computation and redundant calculations, especially if not managed properly. To optimize the recursive solution:

1. Memoization: Store the results of expensive function calls and reuse the cached result when the same inputs occur again. This technique avoids redundant calculations and reduces time complexity to O(n) by storing intermediate results.
2. Iterative Approach: For some problems, converting the recursive approach to an iterative one (using loops) can be more efficient and avoid the overhead associated with recursive function calls and stack space.