**Financial Forecasting**

1. **Concept of Recursion**

Recursion is a programming technique where a function calls itself to solve smaller instances of the same problem. Each recursive call breaks the problem down further until it reaches a base case, which is a condition that stops further recursion.

Recursion helps simplify complex problems by:

* Reducing the amount of code needed.
* Breaking problems into smaller sub-problems, which are easier to solve.
* Avoiding the need for manual loops, especially in problems with branching or nested structures.

In Financial Forecasting , recursion can model the concept of:

**FutureValue(year) = FutureValue(year - 1) × (1 + rate)**

1. **Analysis**

The recursive algorithm calculates the future value of an investment over n years using the formula: FutureValue(n) = FutureValue(n - 1) \* (1 + rate)

Each recursive call reduces the number of years by 1 until it reaches 0.

This creates exactly **n recursive calls**, where n is the number of years.

The recursive algorithm has linear time complexity (O(n)), making it efficient for small to moderate input sizes. However, due to its space complexity of (O(n)), it may not be suitable for large values without optimization.

A simple and effective way to optimize the recursive logic is to replace it with an **iterative approach**, which avoids using the call stack.

For example , the iterative approach code :

public static double forecastIterative(double amount, double rate, int years) {

for (int i = 0; i < years; i++) {

amount \*= (1 + rate); // Compound the amount annually

}

return amount;

}

In my code I have implemented both the approach to compare.