Financial Forecasting Tool using Recursion

# 1. Understand Recursive Algorithms

Recursion is a programming technique where a method calls itself to solve smaller versions of a problem. It helps simplify problems that have repeated or self-similar patterns, such as financial forecasting over multiple years.

Example of a recursive method:

int factorial(int n) {  
 if (n == 1) return 1;  
 return n \* factorial(n - 1);  
}

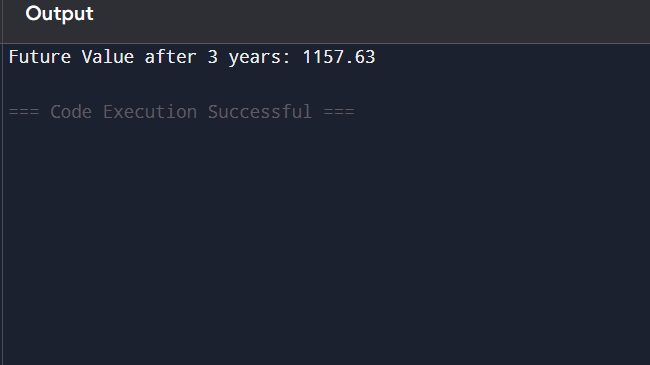
# 2. Setup

We create a method named forecast which accepts the current amount, annual growth rate, and number of years to calculate future value recursively.

# 3. Implementation

Recursive method to calculate future value:

public class Forecast {  
 public static double forecast(double amount, double rate, int years) {  
 if (years == 0) return amount;  
 return forecast(amount \* (1 + rate), rate, years - 1);  
 }  
  
 public static void main(String[] args) {  
 double current = 1000;  
 double growth = 0.05;  
 int futureYears = 3;  
  
 double futureValue = forecast(current, growth, futureYears);  
 System.out.printf("Future Value after %d years: %.2f\n", futureYears, futureValue);  
 }  
}



# 4. Analysis

Time Complexity: O(n), where n is the number of years (each year requires one recursive call).

Problem: For very large values of years, recursion can cause a stack overflow.

To avoid excessive computation, recursion can be replaced by iteration:

public static double forecastIterative(double amount, double rate, int years) {  
 for (int i = 0; i < years; i++) {  
 amount \*= (1 + rate);  
 }  
 return amount;  
}