# Financial Forecasting

## 1. Understand Recursive Algorithms

Recursion is a programming technique where a function calls itself to solve smaller instances of the same problem. It is useful when a problem can be broken down into subproblems of the same type. Recursive algorithms simplify code by replacing loops with function calls. However, they can lead to excessive computation or stack overflow if not properly managed.

## 2. Setup

We define a method to calculate the future value of an investment using a recursive approach. The formula for compound growth is used recursively:

Future Value = Principal × (1 + Rate)^Years

## 3. Implementation

The following Java code calculates the future value of an investment using recursion with memoization to optimize performance:

import java.util.\*;  
  
public class FinancialForecasting {  
  
 private static Map<Integer, Double> memo = new HashMap<>();  
  
 public static double calculateFutureValueMemoized(double principal, double rate, int years) {  
 if (memo.containsKey(years)) {  
 return memo.get(years);  
 }  
  
 if (years == 0) {  
 return principal;  
 }  
  
 double result = calculateFutureValueMemoized(principal \* (1 + rate), rate, years - 1);  
 memo.put(years, result);  
 return result;  
 }  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
  
 System.out.print("Enter the initial amount: ");  
 double principal = scanner.nextDouble();  
  
 System.out.print("Enter the annual growth rate (as a decimal): ");  
 double rate = scanner.nextDouble();  
  
 System.out.print("Enter the number of years: ");  
 int years = scanner.nextInt();  
  
 double futureValue = calculateFutureValueMemoized(principal, rate, years);  
 System.out.printf("Future Value : %.2f%n", futureValue);  
  
 scanner.close();  
 }  
}

## 4. Analysis

Time Complexity:

• Without memoization, the time complexity is O(2^n) as the function recomputes values repeatedly.

• With memoization, the time complexity reduces to O(n), since each year is calculated only once.

Optimization:

Memoization stores previously computed values in a map so that the same computation is not repeated. This drastically improves performance and prevents redundant recursive calls.