Week - 1: Algorithms & Data Structures

**Financial Forecasting**

# Understanding Recursive Algorithms

# Recursion is a method where a function calls itself to solve smaller instances of a problem. It's especially useful for problems that can be broken down into smaller, similar subproblems. In financial forecasting, recursion can simplify the calculation of future values by applying the same growth logic repeatedly.

# Setup

# We define a method that uses recursion to compute the future value of an investment based on:

# The initial amount

# The annual growth rate

# The number of years

1. **Implementation**

import java.util.Scanner;

**//recursive method to calculate future value**

public class FinForecast {

public static double futureValue(double principal, double rate, int years) { //using recursion to calcute future value

if (years == 0) {

return principal;

}

return futureValue(principal, rate, years - 1) \* (1 + rate);

**//using the same method to calculate the future value**

}

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 result = futureValue(principal, rate, years);

System.out.printf("Future Value : %.2f\n", result);

}

}

1. **Analysis**

**Time Complexity:**

* The time complexity of the recursive futureValue() method is O(n), where n is the number of years. For each year, a recursive call is made until the base case is reached.

**Optimization:**

* In this simple recursion, the number of calls grows linearly with years. However, if you had overlapping subproblems (like in Fibonacci), it could lead to exponential time complexity. In such cases, memoization or dynamic programming would help avoid repeated computation.
* For this problem, since each year's value depends directly and only on the previous year's, the recursion is efficient and doesn’t need further optimization.