**Financial Forecasting**

**Recursion :** Recursion is a programming technique where a method calls itself to solve a smaller instance of the same problem. It is particularly useful when a problem can be broken down into similar sub-problems.

**Code:**

public class FinancialForecast {

public static double futureValue(double presentValue, double rate, int years) {

if (years == 0) {

return presentValue;

}

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

}

public static void main(String[] args) {

double presentValue = 1000.0;

double rate = 0.05; // 5% annual growth

int years = 5;

double result = futureValue(presentValue, rate, years);

System.out.println("Future Value after " + years + " years: " + result);

}

}

**Output:**  
  
**Time Complexity:**

* The recursive solution has **O(n)** time complexity because it makes one recursive call per year until n == 0.

**Space Complexity:**

* The space complexity is also **O(n)** due to the call stack, which holds one frame per recursive call.

**Optimization:(Memoization)**public static double futureValueIterative(double presentValue, double rate, int years) {

double result = presentValue;

for (int i = 1; i <= years; i++) {

result \*= (1 + rate);

}

return result;

}  
  
The above optimization prevents stack overflow by optimizing space.