**DATA STRUCTURES AND ALGORITHMS**

**Exercise 7: Financial Forecasting**

Recursion is a concept in programming where a function calls itself directly or indirectly to solve smaller parts of a bigger problem. It continues to do so until it reaches a base case, which is used to stop the recursion process.

Recursion is especially useful for problems that can be broken down into smaller subproblems with similar structure. It simplifies code that would otherwise require complex loops or stacks.

**Method to calculate the future value using a recursive approach:**

import java.util.Scanner;

public class Main {

public static double calcFutureVal(double pv, double r , int n){ //Recursive function to predict future value

if(n == 0){ //Base Case

return pv;

}

return calcFutureVal(pv \* (1+r), r, n - 1);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the present value: ");

double presentValue = sc.nextDouble(); //Get the present value from the user

System.out.println("Enter the interest rate (as decimal value)");

double rate = sc.nextDouble(); // Get the interest rate(as decimal value) from the user

System.out.println("Enter the number of years: ");

int n = sc.nextInt(); // Get the time period from the user

double futureValue = calcFutureVal(presentValue,rate,n);

System.out.println("Future Value: " + futureValue);

}

}



**TIME COMPLEXITY:**

O(N):The method is called once for each value of n down to 0.

So, there are n recursive calls : linear complexity.

**OPTIMIZATION :**

We can optimize this code by using either an iterative approach which will reduce the space complexity to O(1) or we can use java in-built function Math.pow() to achieve 0(1) time complexity but it is not recommended.