**Data Structures and Algorithms – Exercise 7**

**Financial Forecasting Example**

**Recursion**  
**Recursion** is a programming concept where a function **calls itself** to solve smaller parts of a problem until it reaches a base case.

Recursion is especially useful for solving problems that have a **repetitive, nested, or hierarchical structure**, where a large problem can be broken down into **smaller sub problems of the same type**. It allows for **clean, readable,** and often more **intuitive** solutions.

**Main.java**

import java.util.\*;

public class Main{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("\nWelcome to Financial Forecasting.\n");

        System.out.println("Enter the Principal Amount");

        int principalAmount = sc.nextInt();

        System.out.println("Enter the Growth Rate (in %)");

        double growthRate = sc.nextDouble();

        growthRate = growthRate/100;

        System.out.println("Enter the Time Duration(in Years).");

        int time = sc.nextInt();

        System.out.printf("\nThe Future Predicted Value is: %.3f",findFutureValue(principalAmount, growthRate, time));

        sc.close();

    }

    static double findFutureValue(int principalAmount, double growthRate, int time){

        if(time==0)

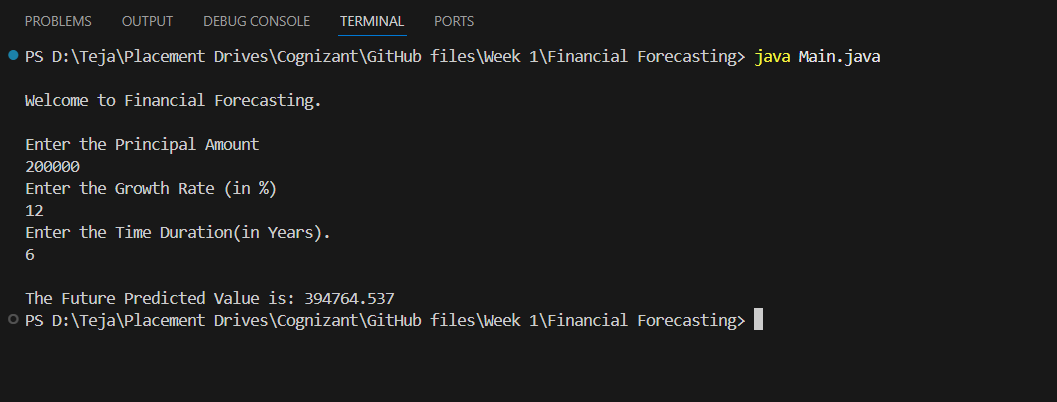
        return principalAmount;

        return findFutureValue(principalAmount, growthRate, time-1)\*(1+growthRate);

    }

}

**Output**

****

**Time Complexity – O (t) where, t is time**

To improve the performance and optimize the program, we can use mathematical computing formula for this calculation. Instead of recursion we can use iterative methods also to reduce the complexity. This also avoids the recursion stack used in recursive method.