**Exercise 7: Financial Forecasting**

**Scenario:**

You are developing a financial forecasting tool that predicts future values based on past data.

**Steps:**

1. **Understand Recursive Algorithms:**
   * Explain the concept of recursion and how it can simplify certain problems.
2. **Setup:**
   * Create a method to calculate the future value using a recursive approach.
3. **Implementation:**
   * Implement a recursive algorithm to predict future values based on past growth rates.
4. **Analysis:**
   * Discuss the time complexity of your recursive algorithm.
   * Explain how to optimize the recursive solution to avoid excessive computation.

SOLUTION:

* Recursion is a method where the function calls itself repeatedly until a base case is satisfied.
* Recursion helps in breaking down complex problems into smaller sub problems making the code less complex to understand.

**FF.java**

**package** forecastTool;

**public** **class** FF {

**public** **static** **double** GrowthRate(**double** principal, **double**[] growthRates) {

**for** (**double** rate : growthRates) {

principal \*= (1 + rate);

}

**return** principal;

}

**public** **static** **double** GrowthRatesRec(**double** principal, **double**[] growthRates, **int** year) {

**if** (year == growthRates.length) {

**return** principal;

}

**return** *GrowthRatesRec*(principal \* (1 + growthRates[year]), growthRates, year + 1);

}

}

**Main.java**

**package** forecastTool;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

**double** principal = 10000;

**double**[] growthRates = {0.05, 0.03, -0.02, 0.04, 0.06};

**long** startIter = System.*nanoTime*();

**double** futureValIter = FF.*GrowthRate*(principal, growthRates);

**long** endIter = System.*nanoTime*();

**long** timeIter = endIter - startIter;

**long** startRec = System.*nanoTime*();

**double** futureValRec = FF.*GrowthRatesRec*(principal, growthRates, 0);

**long** endRec = System.*nanoTime*();

**long** timeRec = endRec - startRec;

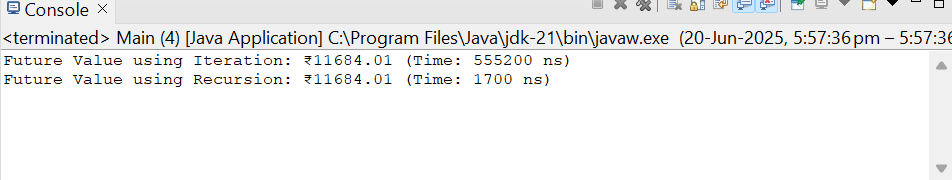
System.***out***.printf("Future Value using Iteration: ₹%.2f (Time: %d ns)%n", futureValIter, timeIter);

System.***out***.printf("Future Value using Recursion: ₹%.2f (Time: %d ns)%n", futureValRec, timeRec);

}

}

**OUTPUT:**

****

* The time complexity of recursion is O(n).
* Can optimize using tail recursion or iterative method for linear sequential priblems.