**Financial Forecasting:**

**Recursion:**

**Recursion** is a technique where a method calls itself to solve a problem by breaking it down into smaller subproblems.

**Code:**

public class FinancialForecasting {

public static double predict(double currentvalue,double

growthrate,int years) {

if(years==0) return currentvalue;

return predict(currentvalue\*(1+growthrate),growthrate,years-1);

}

public static void main(String[] args) {

double currentvalue = 10000;

double growthrate = 0.05;

int years = 3;

double futurevalue = predict(currentvalue,growthrate,years);

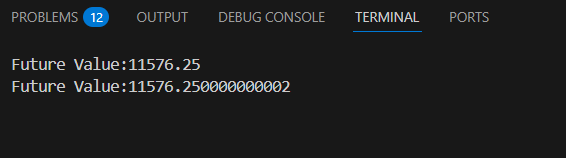
System.out.println("Future Value:"+futurevalue);

System.out.println("Future Value:"+currentvalue\*Math.pow(1+growthrate,years));

}

}

**Output:**

****

**Time Complexity:** Here in this java code the time complexity for recursion is O(n) where the recursion call is happening base on the no of years.

**Optimization Approach:** The optimization approach for this problem is to use mathematical formula where we can get in output in O(1) time complexity which avoid unnecessary recursive calls. The formula used in this problem is: Future Value = Current Value\*(1+growth rate)^years.