

Week 1: Data Structures and Algorithms:

Exercise 7: Financial Forecasting

1: Understanding Recursive Algorithms

Recursion is a programming technique where a function calls itself to solve smaller instances of the same problem.

It's useful for breaking problems into manageable sub-problems, like computing factorials, traversing trees, or predicting financial growth over time (this example).

2: Implementation

The Implementation is straightforward, having a recursive method to calculate future value and a main function to execute the required program

FinancialForecasting.java

```
public class FinancialForecasting {

    //SIMPLE RECURSIVE METHOD TO CALCULATE FUTURE VALUE
    public static double forecast(double cur, double rate, int period) {
        if (period == 0) {
            return cur;
        }
        return forecast(cur, rate, period - 1) * (rate + 1);
    }

    //MAIN METHOD
    public static void main(String[] args) {
        double base_amount = 1000.0;
        double growth_rate = 0.05; // 5%
        int period = 5; //5 YEARS

        double future_value = forecast(base_amount, growth_rate, period);
        System.out.println("FUTURE VALUE AFTER " + period + " YEARS IS : " + future_value
    );
    }
}
```

3: Analysis of Algorithm

This recursive function has $O(n)$ time complexity, where n is the number of periods. It makes a single recursive call per period.

We can further optimize the algorithm by implementing memoization for repeating sub problems. Eg – In case of Fibonacci algorithm,

OUTPUT

```
edetailsInExceptionMessages - cp C:\Users\Manjua\AppData\Local\Temp\jdt_ws\CODE_938f47be\bin FinancialForecasting "
FUTURE VALUE AFTER 5 YEARS IS : 1276.2815625000003
```