

Exercise 7: Financial Forecasting

1. Understanding Recursive Algorithms

What is recursion?

Recursion is when a function calls itself to solve smaller versions of a problem. It's like tackling a big problem by breaking it down into smaller problems repeatedly until we reach the smallest one, called the base case.

Recursion is useful for problems like calculating factorials, generating the Fibonacci series, and sometimes for financial forecasting. However, it's not always the most efficient method.

2. Setup

We will create a simple recursive method to predict the future value of an investment based on a yearly growth rate.

We assume the growth rate remains constant.

3. Implementation

```
public class FinancialForecast {

    // Recursive function to calculate future value
    public static double predictFutureValue(double currentValue, double growthRate, int years) {
        if (years == 0) {
            return currentValue;
        }

        // growthRate is in percentage, so divided by 100
        double nextValue = currentValue * (1 + growthRate / 100);
        return predictFutureValue(nextValue, growthRate, years - 1);
    }

    public static void main(String[] args) {
        double current = 10000;
        double rate = 5; // 5% yearly growth
        int year = 3;

        double future = predictFutureValue(current, rate, year);
        System.out.println("Future value after " + year + " years = " + future);
    }
}
```

OUTPUT:

```
PS C:\Users\user> cd "e:\GenC Hands on\Week 1\java codes\"
}
Future value after 3 years = 11576.25
PS E:\GenC Hands on\Week 1\java codes>
```

4. Analysis

Time Complexity:

The time complexity of this recursive function is $O(n)$, where n is the number of years. The function calls itself once for each year until it reaches 0.

If we forecast for 10 years, there will be 10 recursive calls.

How to optimize?

In this case, optimization is not necessary because we only call the function once per year. However, for other recursive problems like the Fibonacci sequence, we can use a technique called memoization or convert it into an iterative approach to avoid repeating calculations.

We could also rewrite this using a loop instead of recursion. This would be faster and require less memory, especially for a large number of years.