**Financial Forecasting – Explanation & Time Complexity**

**Problem Overview:**

In this exercise, we’re building a **simple forecasting tool** that predicts **future values** based on past data and a growth rate. We’re using **recursion** to calculate values over time repeatedly.

**What is Recursion?**

Recursion means **a function calling itself** to solve smaller subproblems. It’s often used when the problem has a repeating pattern like:

“To calculate year 5, you need year 4;  
To calculate year 4, you need year 3...”

**How We Use It Here:**

Let’s say you have an initial amount and a fixed growth rate (like 10%):

futureValue(year) = futureValue(year - 1) + (growthRate × futureValue(year - 1))

You stop when year = 0 (base case).

**Time Complexity**

| **Recursion Type** | **Time Complexity** |
| --- | --- |
| Basic Recursion | O(n) |
| With Memoization | O(n) |

Without memoization, repeated calls can slow things down.  
If you calculate the same year more than once, it’s inefficient.

**Conclusion:**

Recursion is great for modeling future predictions based on past values. But for large time ranges, it can get slow.  
To make it faster, always try to optimize with **loops** or **memoization**.