# Financial Forecasting Tool Analysis

1. **Understanding Recursive Algorithms**

## What is Recursion

Recursion is when a method calls itself to solve a smaller piece of the problem. It’s like breaking a big task into smaller ones until you hit a simple case that stops the process. In financial forecasting, recursion helps calculate future money by applying a growth rate year by year.

# Setup

## Future Value Method

The tool has a method, calcFutureMoney, that predicts future value. It takes:

* + - money: Starting amount (e.g., $1000).
    - rate: Annual growth rate (e.g., 0.05 for 5%).
    - yrs: Number of years to predict.

It uses recursion to multiply the money by (1 + rate) each year.

# Implementation

## Recursive Algorithm

The calcFutureMoney method works like this:

* + - If yrs is 0, return money (no growth).
    - Otherwise, call calcFutureMoney with yrs - 1, then multiply by (1 + rate). For example, for $1000 at 5% over 2 years:
    - Year 2: Money = (Year 1 money) \* 1.05
    - Year 1: Money = $1000 \* 1.05
    - Year 0: $1000

# Analysis

## Time Complexity

The algorithm takes O(n) time, where n is yrs. It makes one recursive call per year, and each call does a quick multiplication, so total time grows with n. It also uses O(n) space for the call stack.

## Optimizing the Algorithm

The recursive code is clear but can be improved:

* + - **Loop Instead**: Use a for loop to multiply money by (1 + rate) n times. This saves space (O(1)) and avoids stack overflow for big n.
    - **Direct Formula**: Use FV = money \* (1 + rate)*nfor O*(1)*t*ime, butwatchforroundingerrorswithbignu

**Recommendation**: For a real app, use a loop or the formula for speed and re- liability, especially for long forecasts. Recursion is good for learning or small n.