**Exercise 7: Financial Forecasting**

**Understand Recursive Algorithms**

**What is Recursion?**

* **Recursion** is a method where a function calls itself to solve smaller instances of a problem.
* Each recursive call should have:
  + **Base Case** – to stop recursion.
  + **Recursive Case** – where the problem is divided and function calls itself.

**Why Use Recursion?**

* Simpler and more intuitive code for problems like:
  + Factorials
  + Fibonacci sequence
  + Tree traversal
  + Compound interest or forecasting based on growth

**Setup:** Forecasting Formula **(**Concept**)**

We'll forecast future value (FV) using:

FV = PV \* (1 + r)^n

Where:

* PV = present value
* r = growth rate (as decimal, e.g., 0.05 for 5%)
* n = number of years

**Analysis**

**Time Complexity:**

* **Recursive Forecast**:
  + Time: O(n) (calls itself n times)
  + Space: O(n) due to recursive call stack
* **Optimized (Math.pow)**:
  + Time: O(log n) (uses fast exponentiation internally)
  + Space: O(1)