Exercise 7: Financial Forecasting

Understand Recursive Algorithms

What is Recursion?

Recursion is a programming technique where a method **calls itself** to solve smaller instances of a problem.

- Example: f(n) = f(n-1) + something...
- It simplifies problems that have **repetitive, nested, or hierarchical structures**.

Setup :

Define the Forecasting Function:

Forecast the future value of an investment using this formula:

FutureValue(n) = PresentValue * (1 + growthRate)^n.

We can write a recursive function to simulate this exponential growth over n periods (e.g., years).

Analysis :

Time Complexity

- The function calls itself once for each year: O(n)
- For n years, the function will be called n + 1 times (including base case).

Space Complexity

- Due to recursion, the function uses a **call stack**: O(n) space

How to Optimize

1. **Use Iteration** instead of recursion to avoid stack overflow.
2. **Use Memoization** if overlapping subproblems exist (not needed here).
3. **Mathematically simplify**:

Use power function: **presentValue * Math.pow(1 + growthRate, years)** — which is optimal.