What is Recursion?

Recursion is a programming technique where a function calls itself to solve a smaller instance of the same problem.

In simple terms:  
“To solve a big problem, break it into smaller versions of the same problem, solve the smallest one, and build your way back up.”

Every recursive function must have:

1. Base Case – The condition when the function stops calling itself.
2. Recursive Case – The part where the function calls itself with a smaller input.
3. Analysis:

o Discuss the time complexity of your recursive algorithm.

Time Complexity: O(n)

* For each year, the function calls itself once with years - 1.
* This continues until years == 0.
* So, for n years, there will be n recursive calls.

o Explain how to optimize the recursive solution to avoid excessive computation.

a. Use Iteration Instead of Recursion

* Convert the recursive function into a loop-based version.
* Iteration is faster and uses O(1) space.

java

b.Memoization

Memoization is useful when the recursive function recomputes the same values repeatedly (like in Fibonacci). In forecasting, this isn't typically needed unless we want values for many intermediate years.