Amrita Kumari (6362289)

WEEK-1
Data Structure and Algorithm
Exercise 7: Financial Forecasting

Understanding Recursive Algorithm

Recursion is a method where a function calls itself to solve smaller instances of a problem.It is useful when a problem can be broken down into similar sub problems, such as:
* Factorials
* Fibonacci Series
* Tree traversals
* Forecasting patterns

e.g. To predict value in year **n**, you need value of year **n-1**, which depends on **n-2**, and so on until base year.

Analysis

I. Recursive only:
- T(n) = T(n-1) + O(1)
- Time complexity: O(n)
- Space complexity: O(n) (due to function call stack)

II. With saved computations:
- Each value is computed once and stored.
- Time: O(n)
- Space: O(n) (array + call stack)

Recursion is useful for solving problems that can be divided into smaller subproblems, but in its basic form, it can be inefficient due to repeated calculations. In financial forecasting, for instance, calculating the value for year 5 requires computing values for years 4, 3, and so on — often multiple times. This results in unnecessary computations and increased time complexity.

To improve efficiency, memoization is used. It stores the result of each subproblem when it's first computed, so future recursive calls can simply reuse the stored value. This avoids redundant work and reduces the risk of stack overflow in deep recursion. In forecasting, memoization ensures each year's value is calculated just once, making the solution faster and more suitable for larger inputs.