**Exercise 7: Financial Forecasting**

**Explain the concept of recursion and how it can simplify certain problems**

Recursion is a process in which a function calls itself directly or indirectly is called recursion and the corresponding function is called a recursive function. Using a recursive algorithm, certain problems can be solved quite easily.

A recursive function generally has two main components:

* **Base Case:** A condition under which the function stops calling itself, preventing an infinite loop.
* **Recursive Case**: The part of the function where it calls itself with a subset of the original problem.

Recursion can simplify the solution of complex problems by breaking them down into simpler sub-problems of the same type.

Certain types of problems solved using recursion,

* Tower of Hanoi
* Tree traversals

**Discuss the time complexity of your recursive algorithm.**

The time complexity of the recursive calculateFutureValue function is O(n), where n is the number of years. This is because the function makes n recursive calls before reaching the base case.

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

* **Memorization**: To avoid recalculating the same values multiple times, we could use memorization, storing the results of previous calculations. In the implemented recursive function, each recursive call depends on the result of the previous year only, and thus memoization might not provide significant benefits unless there are overlapping subproblems.
* **Iterative Approach**: Another way to optimize the solution is to use an iterative approach instead of recursion. This can be more efficient in terms of space complexity since recursion uses the call stack to keep track of function calls, leading to O(n) space complexity. An iterative approach would use O(1) space complexity.