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

Recursion is a technique where a function calls itself to solve a problem. It simplifies complex problems by breaking them down into smaller, more manageable sub-problems.

Characteristics of Recursive Algorithms:

* Base Case: A condition that stops the recursion.
* Recursive Case: The part where the function calls itself with a modified argument.

Example Problem: Factorial

* Recursive Definition: factorial(n) = n \* factorial(n-1) with factorial(0) = 1.

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

The time complexity of the recursive algorithm is O(n), where n is the number of years. This is because each recursive call reduces the problem by one year, leading to n recursive calls**.**

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

One way to optimize a recursive solution is by using **memoization** to store previously computed results and avoid redundant calculations. In this case, memoization is not necessary as each calculation is unique to the given year and does not overlap. However, for more complex recursive problems, memoization can significantly reduce computation time.

Another approach to optimize is converting the recursive solution to an **iterative solution** to avoid the overhead of recursive calls.