Q7. Understand Recursive Algorithms:

Recursion is a programming technique where a function calls itself to solve a problem by breaking it down into smaller, similar sub-problems. It consists of two main parts:

*Base case*: The simplest form of the problem that can be solved directly.

*Recursive case*: The general case where the problem is broken down and the function calls itself with a simpler version of the problem.

Recursion can simplify certain problems by making the code more elegant and easier to understand, especially for problems that have a natural recursive structure.

Analysis:

*Time complexity*:

* Without optimization: O(n), where n is the number of years.
* With memoization: O(n) for the first calculation, but O(1) for subsequent calculations of previously computed values.

*Optimization*:

We've used memoization to optimize the recursive solution. Memoization is a technique where we store the results of expensive function calls and return the cached result when the same inputs occur again. This avoids redundant calculations and can significantly improve performance, especially for larger values of n.