**Understand Recursive Algorithms:**

Recursion is a programming technique where a method calls itself to solve a problem. Each recursive call should reduce the problem into a smaller version, eventually reaching a base case, which ends the recursion.

**Time Complexity:**

Time complexity can be calculated using techniques like substitution or master’s theorem but best way to do it is by recursive tree method.

The time complexity of the given recursive algorithm of the particular question is:

O(n) : since we perform one recursive call for each year (until we reach the base case).

**Optimization Technique:**

To avoid excessive computation in a recursive solution, you can use **memoization**, which stores the results of previous recursive calls so they don’t need to be recomputed. This reduces the number of function calls and improves performance, especially in problems with overlapping subproblems (like Fibonacci or dynamic programming).