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

Recursion is a programming technique where a function calls itself to solve a smaller instance of the same problem. It simplifies problems that have a natural hierarchical or repetitive structure, like tree traversal, factorial calculation, or solving the Tower of Hanoi, by reducing complex tasks into base and recursive cases.

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

The recursive findRecursive() algorithm has a time complexity of O(n), where *n* represents the number of years. This is due to the function invoking itself once for each year, leading to a total of *n* recursive calls throughout the execution.

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

To optimize a recursive solution and avoid excessive computation, use memoization or dynamic programming. Memoization stores results of previously solved subproblems to prevent redundant calculations. This reduces time complexity, especially in problems with overlapping subproblems. The algorithm findRecursiveMemo()

