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

Concept of Recursion:

Recursion is a programming technique where a function calls itself to solve a problem. The function typically breaks the problem down into smaller, more manageable sub-problems, which are then solved recursively. Recursion can simplify problems that have a natural hierarchical structure, such as tree traversals, sorting algorithms, and mathematical sequences.

A recursive function typically has two main parts:

Base Case: A condition under which the function returns a result directly without further recursion. This prevents infinite recursion.

Recursive Case: The part of the function where it calls itself with a smaller or simpler version of the original problem.

**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 periods. This is because the function makes a recursive call for each period, leading to n recursive calls in total.

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

To optimize the recursive solution and avoid excessive computation, especially when dealing with large values of n, we can use memorization or dynamic programming. This involves storing the results of intermediate calculations and reusing them, rather than recalculating the values in each recursive call.