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

Recursion is a programming technique where a function calls itself to solve smaller instances of the same problem. It simplifies problems by breaking them down into smaller, more manageable sub-problems. Recursion is particularly useful for problems that have a natural hierarchical structure, such as tree traversals, factorial calculations, and certain types of dynamic programming

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 until it reaches the base case.

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

To avoid excessive computation and potential stack overflow issues with deep recursion, we can use memoization or iterative approaches. Memoization involves storing the results of expensive function calls and reusing them when the same inputs occur again. Alternatively, an iterative approach can be used to avoid recursion altogether.