**Understand Recursive Algorithms:**

**Q1. Explain the concept of recursion and how it can simplify certain problems.**

Answer: Recursion is a programming technique where a method calls itself repeatedly until it reaches a base case that stops the recursion.

Recursion can simplify certain problems by breaking them down into smaller sub-problems of the same type, which can be solved independently.

**Analysis:**

**Q1. Discuss the time complexity of your recursive algorithm.**

Answer: The time complexity of this recursive algorithm is O(n), where n is the number of periods. This is because each recursive call reduces the number of periods by 1, until the base case is reached.

**Q2. Explain how to optimize the recursive solution to avoid excessive computation.**

Answer: This recursive solution can lead to excessive computation if the number of periods is large, since each recursive call creates a new stack frame. To optimize this solution, we can use a technique called memoization, which stores the results of expensive function calls and returns the cached result when the same inputs occur again.

This optimized solution has a time complexity of O(n), but with a much smaller constant factor, since we avoid redundant computations by storing the results in a cache.