# **Analysis:**

## Discuss the time complexity of your recursive algorithm

The time complexity of the recursive algorithm for calculating the future value can be analyzed based on the number of recursive calls made by the function.

Time Complexity Analysis

1. Recursive Function Calls:
   * The function ‘calculateFutureValue’ is called recursively once for each period until the base case is reached. Specifically, if the number of periods is n, the function will call itself n times.
2. Base Case:
   * The base case occurs when the periods parameter reaches 0, at which point the function returns the present value without making further calls.
3. Overall Complexity:
   * Since each call involves a constant amount of work (multiplication and addition), the time complexity of the recursive function is linear with respect to the number of periods.

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

To optimize the recursive solution for calculating future values and avoid excessive computation, we can use the following techniques:

1. Memoization:

Memoization is a technique where we store the results of expensive function calls and reuse them when the same inputs occur again. This can significantly reduce the number of recursive calls.

2. Iterative Approach:

Alternatively, you can replace the recursive method with an iterative approach, which avoids the overhead of function calls entirely:

Implementation:

* Use a loop to calculate the future value, which allows you to maintain a constant space complexity O(1).