1. **Recursion** is a programming technique where a function calls itself in order to solve a problem. Recursive algorithms can simplify problems by breaking them down into smaller, more manageable sub-problems. Each recursive call should bring the problem closer to a base case, which terminates the recursion.

For example, calculating the factorial of a number n (i.e., n!) can be simplified using recursion:

n! = n × (n−1)!

The base case here is 0! = 1

4. **Time Complexity:**

The time complexity of the recursive algorithm is O(n), where n is the number of periods. This is because the algorithm makes one recursive call per period.

**Optimizing the Recursive Solution:**

Recursive algorithms can be optimized using **memoization** or **dynamic programming** to avoid excessive recomputation. Memoization involves storing the results of expensive function calls and reusing them when the same inputs occur again.