1. **Concept of Recursion**

**Recursion**: A technique where a function calls itself in order to solve a problem. It often breaks a problem into smaller, more manageable sub-problems.

**Base Case**: The condition under which the recursion stops.

**Recursive Case**: The part of the function that reduces the problem and makes a recursive call.

Simplifies the code for problems that have a natural recursive structure.

4a.) **Time Complexity**: O(n), where n is the number of periods. Each recursive call processes one period, leading to a linear time complexity.

4b.) **Optimization**

* **Memoization**: To avoid redundant calculations, store results of sub-problems.
* **Iterative Approach**: Convert the recursive solution to an iterative one if recursion depth becomes an issue. This avoids stack overflow and may be more efficient in terms of space.