## **Recursive Algorithms**

Recursion is a technique where a function calls itself to solve a problem. It's particularly useful when a problem can be broken down into smaller, similar subproblems.

### **Analysis**

* **Time complexity:** O(n), where n is the number of periods. Each recursive call reduces the problem size by 1, leading to n recursive calls in total.

## **Optimizing the Recursive Solution**

While the recursive solution is straightforward, it can be inefficient for large values of periods due to excessive function calls and stack usage.

**Iterative approach** is a more efficient alternative.

**Memoization:** For more complex recursive functions, memoization can be used to store intermediate results and avoid redundant calculations.

**Tail Recursion:** Some compilers optimize tail-recursive functions into iterative code, potentially improving performance. However, this is not guaranteed across all compilers.