**ANSWERS:**

**What is Recursion?**

Recursion is a is a programming technique where a function **calls itself** to solve smaller instances of the same problem until a **base case** is reached or it’s a technique where a method calls itself to solve a problem by **breaking it down into smaller subproblems**.

**Example:** Calculating factorial:  
factorial(n) = n \* factorial(n - 1)[One of the common problem solved by recursion function]

Recursion is especially useful for problems that:

* Have a **repetitive or self-similar structure**
* Can be broken into **subproblems of the same type**
* Are easier to define in terms of **previous results.**

ANALYSIS:-

**Time Complexity**

* **Time Complexity:** O(n) – since each year results in one recursive call
* **Space Complexity:** O(n) – due to function call stack (not ideal for large n)

**🔹 Optimization Techniques**

1. **Memoization :**
   * Store results of each recursive call to avoid recomputation.
   * Useful if future value depends on more than one prior result (like Fibonacci series).