 **Understand Recursive Algorithms:**

* **Recursion** is a method where a function calls itself to solve smaller instances of the same problem. It simplifies complex problems by breaking them down into more manageable sub-problems.
* Recursive algorithms can be particularly useful for problems that have a natural recursive structure, such as tree traversals or the calculation of factorials.

 **Analysis:**

* **Time Complexity:** The time complexity of a naive recursive approach can be exponential in some cases (e.g., calculating Fibonacci numbers), as each function call generates multiple subsequent calls.
* **Optimization:** To avoid excessive computation, use **memoization** or **dynamic programming** to store intermediate results and reduce redundant calculations.