1. **What is Recursion ?**

When a function calls itself again and again in its own body of code till it reaches a base condition or reaches a desired results then it is called as Recursion .

1. **how it can simplify certain problems ?**

Recursion breaks larger problems into subproblems and then finally combine the solution of each subproblem to achieve a desired results.

**3. Usecases**

* It is useful in divide and conquer problems,Tree traversal and in Mathematical computation
* Simplifies problems with repeated structure (like computing compound interest, Fibonacci, etc.)

**Analysis:**

#### **Time Complexity:**

* Each call reduces years by 1, until 0
* So, Time Complexity = O(n) for n years
* Space Complexity = O(n) (due to recursive call stack)

**Optimization**

**Problem with recursion:**

* Inefficient for large n due to call stack overhead
* No overlapping subproblems here, so memoization is not going to be helpful

**Better Approach: we can use iteration to avoid stack usage:**

public static double forecastValue (double principal, double rate, int years) {

for (int i = 0; i < years; i++) {

principal \*= (1 + rate);

}

return principal;

}

**Time Complexity :** remains O(n),

**space Complexity :** O(1) (much more efficient for large n).