Understand Recursive Algorithms

1. Explain the concept of recursion and how it can simplify certain problems.

Recursion is a programming concept in which a function calls itself to solve a smaller version of the same problem it continues to do so until it reaches a base case (stopping condition).

Recursion helps simplify certain problems by breaking them into **smaller subproblems**. For example, in **Rat in a Maze** problem we explore all the possible paths and choose the best path among them without recursion the logic would be much more complex.

Analysis

1. Discuss the time complexity of your recursive algorithm.

Since the function predictVal() keeps calling itself until years becomes zero and we reduce years by 1, we take n recursive calls when years = n, hence the time complexity is O(n).

2. Explain how to optimize the recursive solution to avoid excessive computation.

To optimize the recursive solutions, we can use –

1. Iteration instead of recursion
2. Use direct mathematical formula (FV = PV*(1+rate)^years)
3. Memoization