Dynamic Programming (DP)

Dynamic Programming is an algorithmic technique used to solve complex problems by breaking them down into smaller subproblems, solving each subproblem only once, and storing the solutions to subproblems to avoid redundant computation.

Types of Dynamic Programming Approaches:

1.Top-Down (Memoization):

Solve the problem recursively, and store the results of subproblems to avoid recomputation.

2.Bottom-Up (Tabulation):

Solve smaller subproblems first and use their results to build up to the solution of the larger problem.

EXAMPLES:

- Fibonacci Sequence
- Longest Common Subsequence (LCS)
- Edit Distance
- Matrix Chain Multiplication

TIME COMPLEXITY AND SPACE COMPLEXITY

1. Fibonacci Series

- Time Complexity: O(n)
- Space Complexity: O(n)
- Data Structure: Array

2. Longest Common Subsequence (LCS)

  - Time Complexity: O(m*n)

  - Space Complexity: O(m*n)

  - Data Structure: 2D Array

3. Knapsack Problem

  - Time Complexity: O(n*W)

  - Space Complexity: O(n*W)

  - Data Structure: 2D Array

4. Shortest Path Problems

  - Time Complexity: O(V^2)

  - Space Complexity: O(V^2)

  - Data Structure: Adjacency Matrix

5. Matrix Chain Multiplication

  - Time Complexity: O(n^3)

  - Space Complexity: O(n^2)

  - Data Structure: 2D Array

6. Edit Distance

  - Time Complexity: O(m*n)

  - Space Complexity: O(m*n)

  - Data Structure: 2D Array