■ Dynamic Programming Patterns – Poster Style Cheat Sheet

1■■ 1D DP (Linear / Sequential)

■ **Idea:** Used when current state depends on last few states — like Fibonacci, stairs, robbery.
■ **Formula:** Formula: dp[i] = f(dp[i-1], dp[i-2], …)
■ **Examples:** Examples: Climbing Stairs (#70), House Robber (#198), Min Cost Climbing Stairs (#746)

2■■ 2D Grid DP

■ **Idea:** Used for grid traversal where transitions depend on up, left, or diagonal cells.
■ **Formula:** Formula: dp[i][j] = grid[i][j] + min(dp[i-1][j], dp[i][j-1])
■ **Examples:** Examples: Unique Paths (#62), Minimum Path Sum (#64)

3■■ Knapsack / Subset DP

■ **Idea:** Used for problems involving inclusion/exclusion choices to reach a target.
■ **Formula:** Formula: dp[i][t] = dp[i-1][t] || dp[i-1][t-arr[i]]
■ **Examples:** Examples: 0/1 Knapsack, Partition Equal Subset Sum (#416), Target Sum (#494)

4■■ Unbounded Knapsack / Coin Change

■ **Idea:** Used when elements can be reused multiple times (unlimited supply).
■ **Formula:** Formula: dp[i][t] = dp[i-1][t] + dp[i][t-arr[i]]
■ **Examples:** Examples: Coin Change (#322), Coin Change II (#518), Combination Sum (#39)

5■■ String DP (LCS / Edit / Palindrome)

■ **Idea:** Used for comparing or modifying strings or substrings.
■ **Formula:** Formula: if(s1[i]==s2[j]) dp[i][j]=1+dp[i-1][j-1]; else dp[i][j]=max(dp[i-1][j],dp[i][j-1])
■ **Examples:** Examples: Edit Distance (#72), LCS (#1143), Longest Palindromic Subsequence (#516)

6■■ Interval DP

■ **Idea:** Used when the problem depends on solving subintervals [i..j].
■ **Formula:** Formula: dp[i][j] = min(dp[i][k] + dp[k+1][j] + cost)
■ **Examples:** Examples: Burst Balloons (#312), Matrix Chain Multiplication, Palindrome Partitioning II (#132)

7■■ Tree DP / DFS + DP

■ **Idea:** Used for tree problems where results from children combine into the parent.
■ **Formula:** Formula: dp[node] = combine(dp[child1], dp[child2], …)
■ **Examples:** Examples: House Robber III (#337), Diameter of Binary Tree (#543), Max Path Sum (#124)

8■■ Bitmask DP (Subset Compression)

■ **Idea:** Used when subsets are represented efficiently as bits (e.g., TSP).
■ **Formula:** Formula: dp[mask][i] = min(dp[mask^(1<<i)][j] + dist[j][i])
■ **Examples:** Examples: TSP, Shortest Path Visiting All Nodes (#847)

9■■ Digit DP

■ **Idea:** Used to count numbers under certain digit constraints.

■ **Formula:** Formula: dp[pos][tight][sum]

■ **Examples:** Examples: Count Numbers with Sum Constraints, Non-negative Integers Without Consecutive Ones

■ State Compression / Multi-State DP

■ **Idea:** Used when multiple dimensions or states interact (like color + bitmask).

■ **Formula:** Formula: dp[state1][state2] = …

■ **Examples:** Examples: Paint N×3 Grid (#1411), TSP Variants