Dynamic Programming Patterns (with C++ Snippets)

1D DP

```cpp
vector<int> dp(n+1, 0);
dp[0] = base_case;
for (int i = 1; i <= n; i++) {
    dp[i] = function_of(dp[i-1], dp[i-2], ...);
}
```

Practice Problems:

- Leetcode 70 - Climbing Stairs

- Leetcode 198 - House Robber

- Leetcode 746 - Min Cost Climbing Stairs

- GFG - Friends Pairing Problem

- Leetcode 91 - Decode Ways

2D Grid DP

```cpp
vector<vector<int>> dp(m, vector<int>(n, 0));
dp[0][0] = grid[0][0];  // or 1
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        if (i > 0) dp[i][j] = min(dp[i][j], dp[i-1][j] + grid[i][j]);
        if (j > 0) dp[i][j] = min(dp[i][j], dp[i][j-1] + grid[i][j]);
    }
}
```

Practice Problems:

- Leetcode 64 - Minimum Path Sum

- Leetcode 62 - Unique Paths

- Leetcode 63 - Unique Paths II

- GFG - Gold Mine Problem

- Leetcode 221 - Maximal Square