Algorithm: Longest Common Subsequence (LCS)

Princess Elara needed to find the longest common pattern between two ancient sea charts to locate hidden treasures. She used the Longest Common Subsequence algorithm to find the common pattern.

#### Initialize Data Structures:

* Princess Elara used a mystical scroll (2D array) to keep track of common patterns.

#### Find Common Pattern:

* She compared the charts and updated the scroll with the length of the longest common subsequence.

#### Implementation:

| **def** lcs(chart1: str, chart2: str) -> int:  m, n = len(chart1), len(chart2)  dp = [[0] \* (n + 1) **for** \_ **in** range(m + 1)]  **for** i **in** range(1, m + 1):  **for** j **in** range(1, n + 1):  **if** chart1[i - 1] == chart2[j - 1]:  dp[i][j] = dp[i - 1][j - 1] + 1  **else**:  dp[i][j] = max(dp[i - 1][j], dp[i][j - 1])  **return** dp[m][n]  *# Example usage:*  chart1 = "AGGTAB"  chart2 = "GXTXAYB"  print(lcs(chart1, chart2)) *# Output: 4 (Common pattern: GTAB)* |
| --- |

#### Explanation:

Initialize:

* dp: A mystical scroll to track common patterns.

Find Common Pattern:

### Princess Elara compared the charts and updated the scroll with the length of the longest common subsequence.