Report

Time Complexity Overview:

The algorithm, designed for global sequence alignment through dynamic programming, demonstrates a time complexity of O(n * m), where N and M denote the lengths of the input sequences.

Score Matrix Initialization:

Time complexity: O(n * m)
The creation of a 2D score matrix (with dimensions (n + 1) x (m + 1)) involves initializing entries to zeros.

Score Matrix Population:

Time complexity: O(n * m)
Filling in the score matrix is executed by considering each cell individually, calculating scores based on match, delete, and insert operations.

Traceback for Alignment Reconstruction:

Time complexity: O(n + m)
In the traceback process, starting from the matrix's bottom-right corner, the algorithm reconstructs the optimal alignment. The traversal encompasses both 'n' and 'm'
elements in the worst-case scenario.

-Overall Time Complexity:

The overall time complexity of the algorithm is O(n * m).

-Explanation of my Approach:

The algorithm is a dynamic programming solution nature accommodates the evaluation of all conceivable alignments, ultimately selecting the one with the highest score.It proves practical for relatively small sequences, finding widespread application in bioinformatics for diverse sequence alignment tasks, establishes a score matrix where each cell signifies the maximum score for aligning prefixes of input sequences 'x' and 'y'. Three fundamental operations (match, delete, and insert) are considered at each matrix cell.

1. Initialization Phase:

- T The lengths of the input sequences **x** and **y** are determined using **len(x)** and **len(y)**.
- A 2D matrix, **score_matrix**, is initialized with dimensions (n + 1) x (m + 1), where n and m are the lengths of **x** and **y**, respectively.
- The initial values in the matrix are set to zeros.

.

2. Score Matrix Computation:

- Nested loops iterate over the cells of the **score_matrix** to compute scores for each position.
- Scores are calculated based on three operations: match, delete, and insert.
- The maximum score among these three operations is assigned to the current cell in the matrix

3. Alignment Reconstruction through Traceback:

- Two lists, **alignment_x** and **alignment_y**, are used to store the aligned characters of sequences **x** and **y**.
- The traceback process starts from the bottom-right cell of the matrix.
- During the traceback, characters are added to the alignment lists based on the comparison of scores and the chosen operation (match, delete, or insert).

4. Final Outcome:

- T The aligned sequences are reconstructed by reversing the lists and joining the characters.
- The final result is printed, displaying the aligned sequences.

Moamen Ahmed Labib 10000369