

The time complexity of the sequence alignment algorithm is  $O(m * n)$ , where  $m$  and  $n$  are the lengths of the input sequences. This complexity arises from the nested loops used to fill the dynamic programming (DP) table and the traceback process.

1. Initialization ( $O(m + n)$ ):

- Filling the first row and column of the DP table requires  $O(m + n)$  operations.

2. DP Table Filling ( $O(m * n)$ ):

- The nested loops fill the DP table based on the recurrence relation, considering three possible operations (match, delete, and insert) at each cell.

3. Traceback ( $O(m + n)$ ):

- The traceback process involves reconstructing the alignment by backtracking through the DP table. This requires  $O(m + n)$  operations.

Overall, the dominant factor in the time complexity is the filling of the DP table.

Approach Explanation:

The sequence alignment algorithm employs a dynamic programming approach to find the optimal alignment between two sequences. Here's a brief overview of the approach:

1. Initialization:

- Initialize a DP table with dimensions  $(n+1) \times (m+1)$ , where  $n$  and  $m$  are the lengths of the input sequences  $x$  and  $y$ .

2. DP Table Filling:

- Fill the DP table by considering three possible operations at each cell: match (diagonal movement), delete (upward movement), and insert (leftward movement).
- The DP table stores the cumulative scores at each position, representing the optimal alignment score up to that point.

3. Traceback:

- Reconstruct the optimal alignment by backtracking through the DP table from the bottom-right to the top-left corner.
- At each step, determine the operation (match, delete, or insert) that led to the current cell's score and append the corresponding characters to the aligned sequences.

4. Output:

- Return the optimal alignment, including the aligned sequences and the alignment score.

The scoring matrix defines the match, mismatch, gap opening, and gap extension penalties, influencing the alignment score. The algorithm maximizes the alignment score by choosing the optimal combination of operations at each step.