

Marwan Walid 7000327

#### Time Analysis:

The provided dynamic programming algorithm for gene sequence alignment has a time complexity of  $O(mn)$ , where  $m$  and  $n$  are the lengths of the input sequences. This efficiency is achieved through a bottom-up dynamic programming approach, which fills a 2D table to store intermediate results. The nested loops iterate through the entire table, and each cell's value is computed in constant time. Therefore, the overall time complexity is linear with respect to the lengths of the input sequences.

#### Approach Explanation:

The algorithm employs a dynamic programming strategy to find the highest-scoring alignment between two gene sequences, considering a given scoring matrix. The approach involves constructing a 2D table (dp) to store the maximum alignment scores for various subproblems. The initialization sets up the base cases for gaps in the sequences.

The dynamic programming table is filled iteratively by considering three possible operations at each step: match, delete, and insert. The algorithm maximizes the score by choosing the operation that yields the highest alignment score.

After calculating the dynamic programming table, the `getAlignment` method traces back the optimal alignment by considering the values in the table. The resulting aligned sequences are then printed.

Additionally, a method named `calculateAlignmentScore` has been added to calculate the total score of the obtained alignment based on the provided scoring matrix.

This approach ensures efficiency by avoiding redundant computations through the use of dynamic programming and guarantees the optimal alignment score. The provided time complexity of  $O(mn)$  makes the algorithm practical for aligning gene sequences of varying lengths.