**Report**

The time complexity of the algorithm I made was O(mn) since only 2 loops were used to create the scoring matrix and two loops were used to trace and get the solution.

My approach to this problem was to first fill the scoring matrix that is used to get the best cost. The first row and column are filled first by adding the penalty for the gap with each letter. Then the rest of the cells are filled with the best possible cost which is either the cost of adding gap in first sequence or in the second sequence or the cost of matching/unmatching the letters. While filling the scoring matrix, another matrix called tracing matrix is being used to store the direction where the current cost was calculated from which are either up, left or diagonal. Then at the end we can trace the path from the last cell in the matrix to form the string sequences from the end. If the direction was up we add a gap in the first sequence and the letter in second sequence. If it was left we do the opposite. If the direction was diagonal we add the letter to first sequence and the letter to the second sequence, they could be matched or unmatched. We keep tracing until reaching the first position in the matrix and by then both sequences are formed.