Resequencing a set of strings based on a target string.

Summary: Given a set \( S = \{S_1, S_2, \ldots, S_l\} \) of \( l \) strings, a text \( T \), and a natural number \( k \), find a string \( M \), which is a concatenation of \( k \) strings (not necessarily distinct, i.e., a string in \( S \) may occur more than once in \( M \)) from \( S \), whose longest common subsequence with \( T \) is largest, where a string in \( S \) may occur more than once in \( M \). Such a string is called a \( k \)-inlay. The resequencing longest common subsequence problem (resequencing LCS problem for short) is to find a \( k \)-inlay for each query with parameter \( k \) after \( T \) and \( S \) are given. In this paper, we propose an algorithm for solving this problem which takes \( O(nml) \) preprocessing time and \( O(\vartheta_k k) \) query time for each query with parameter \( k \), where \( n \) is the length of \( T \), \( m \) is the maximal length of strings in \( S \), and \( \vartheta_k \) is the length of the longest common subsequence between a \( k \)-inlay and \( T \).

Keywords: dynamic programming; longest common subsequences; resequencing; inverted indexing; totally monotone matrices