Journal of Mathematics and Informatics
Vol. 27, 2024, 27-37
ISSN: 2349-0632 (P), 2349-0640 (online)
Published 11 December 2024
www.researchmathsci.org
DOI:http://dx.doi.org/10.22457/jmi.v27a03246

Journal of
Mathematics and Informatics

An Algorithm for the Longest Common Subsequence and Substring Problem for Multiple Strings

Rao Li

Department of Computer Science, Engineering, and Mathematics,
University of South Carolina Aiken, Aiken, SC 29801, USA,
E-mail: raol@usca.edu

Received 2 November 2024; accepted 10 December 2024

Abstract. Let X_1, X_2, ..., X_s and Y_1, Y_2, ..., Y_t be strings over an alphabet ∑, where s and t are positive integers. The longest common subsequence and substring problem for multiple strings X_1, X_2, ..., X_s and Y_1, Y_2, ..., Y_t is to find the longest string which is a subsequence of X_1, X_2, ..., X_s and a substring of Y_1, Y_2, ..., Y_t. In this paper, we propose an algorithm to solve the problem.

Keywords: Algorithm, the longest common subsequence and substring problem, the longest common subsequence and substring problem for multiple strings.

AMS Mathematics Subject Classification (2020): 68W32, 68W40

1. Introduction

Let ∑ be an alphabet and S a string over ∑. A subsequence of a string S is obtained by deleting zero or more elements of S. A substring of a string S is a subsequence of S consists of consecutive elements in S. We say a string is empty if it does not have any element in it. An empty string is a subsequence and a substring of any string. Let X and Y be two strings over an alphabet ∑. The longest common subsequence (resp. substring) problem for X and Y is to find the longest string which is a subsequence (resp. substring) of both X and Y. Both the longest common subsequence problem and the longest common substring problem have applications in different fields. For example, in molecular biology, the lengths of the longest common subsequence and the longest common substring can be used to measure the similarity between two biological sequences. The two problems have been well-investigated in the last several decades. More details on the research of the first problem can be found in [1], [2], [3], [4], [5], [8], [9], [10], [12], [13] and references therein and the second problem can be found in [6], [7], [14] and references therein. Motivated by the two problems above, Li, Deka, and Deka [11] introduced the longest common subsequence and substring problem for two strings X and Y which is to find the longest string such that it is a subsequence of X and a substring of Y. They also proposed an algorithm to solve this problem in [11]. In this paper, we introduce the longest common subsequence and substring problem for multiple strings which is a generalization of the longest common subsequence and substring problem for two stings. Suppose $X_{1,}$ X_2, ..., X_s and Y_1, Y_2, ..., Y_t are strings over an alphabet ∑, where s and t are positive integers. The (s, t)-longest common subsequence and substring problem for multiple strings $X_1, X_2, ..., X_s$ and $Y_1, Y_2, ..., Y_t$ is to find the longest string, denoted $Z(X_1, X_2, ..., X_s; Y_1, Y_2, ..., Y_t)$, which is a subsequence of $X_1, X_2, ..., X_s$ and a substring of $Y_1, Y_2, ..., Y_t$. If Z(X_1, X_2, ..., X_s; Y_1, Y_2, ..., Y_t) does not exist, we say $Z(X_1, X_2, ..., X_s; Y_1, Y_2, ..., Y_t)$ is an empty string. We propose an algorithm to solve the (s, t)-longest common subsequence and substring problem.

2. The preparations for the algorithm

Our algorithm is based on several claims to be proved in this section. Before proving the claims, we need some notations as follows. For a given string $H = h_1 h_2 ... h_k$ over an alphabet ∑, the size of H, denoted |H|, is defined as the number of elements in H. The length of an empty string is zero. The jth suffix of H is the string of $h_j h_{j+1} ... h_k$, where $1 \leq j \leq k$. The ith prefix of H is defined as $H[i] = h_1 h_2 ... h_i$, where $1 \leq i \leq k$. Conventionally, H[0] is defined as an empty string.

Let $X_p = x[p, 1]x[p, 2] ... x[p, m_p]$, where x[p, a] with p is an integer such that $1 \leq p \leq s$ and $1 \leq a \leq m_p$ are elements in an alphabet ∑, be s strings, and $Y_q = y[q, 1]y[q, 2] ...$ y[q, n_q], where y[q, b] with q is an integer such that $1 \leq q \leq t$ and $1 \leq b \leq n_q$ are elements in the alphabet ∑, be t strings. We define $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ as a string satisfying the following conditions, where $1 \leq i_u \leq m_u$ with $1 \leq u \leq s$ and $1 \leq j_v \leq n_v$ with $1 \leq v \leq t$.

(1.1) It is a subsequence of $X_1[i_1] = x[1, 1]x[1, 2] ... x[1, i_1]$.
(1.2) It is a subsequence of $X_2[i_2] = x[2, 1]x[2, 2] ... x[2, i_2]$.
....
(1.s) It is a subsequence of $X_s[i_s] = x[s, 1]x[s, 2] ... x[s, i_s]$.
(2.1) It is a suffix of $Y_1[j_1] = y[1, 1]y[1, 2] ... y[1, j_1]$.
(2.2) It is a suffix of $Y_2[j_2] = y[2, 1]y[2, 2] ... y[2, j_2]$.
....
(2, t) It is a suffix of $Y_t[j_t] = y[t, 1]y[t, 2] ... y[t, j_t]$.
(3.1) Under the conditions above, its length is as large as possible.

Claim 1. If $y[1, j_1], y[2, j_2], ..., y[t, j_t]$ are not the same, then $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ does not exist. Namely, $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ is an empty string.

Proof of Claim 1. Suppose, to the contrary, that $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ exists. Then $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ is not empty. Thus the last element in it must be equal to each of $y[1, j_1], y[2, j_2], ..., y[t, j_t]$ and therefore $y[1, j_1], y[2, j_2], ..., y[t, j_t]$ are the same, a contradiction. Hence $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ does not exist.

Hence the proof of Claim 1 is complete.

Claim 2. Suppose that $u_1 := y[1, j_1] = y[2, j_2] = \cdots = y[t, j_t]$. If $u_1 = x[1, i_1] = x[2, i_2] = \cdots = x[s, i_s]$, then

$$|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| = |Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1 - 1, j_2 - 1, ..., j_t - 1]| + 1.$$

Proof of Claim 2. Our proof is divided into two cases.

Case 1. $Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1 - 1, j_2 - 1, ..., j_t - 1]$ is not empty.

Notice that $Z_\alpha := Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1 - 1, j_2 - 1, ..., j_t - 1]$ is a subsequence of

$X_1[i_1 - 1] := x[1, 1]x[1, 2] ... x[1, i_1 - 1]$,
$X_2[i_2 - 1] := x[2, 1]x[2, 2] ... x[2, i_2 - 1]$,

$$\ldots\ldots\ldots\ldots$$
$$X_s[i_s - 1] := x[s, 1]x[s, 2] \ldots x[s, i_s - 1],$$
and a suffix of
$$Y_1[j_1 - 1] := y[1, 1]y[1, 2] \ldots y[1, j_1 - 1],$$
$$Y_2[j_2 - 1] := y[2, 1]y[2, 2] \ldots y[2, j_2 - 1],$$
$$\ldots\ldots\ldots\ldots$$
$$Y_t[j_t - 1] := y[t, 1]y[t, 2] \ldots y[t, j_t - 1].$$
Since $x[1, i_1] = x[2, i_2] = \cdots = x[s, i_s] = u_1 = y[1, j_1] = y[2, j_2] = \cdots = y[t, j_t]$, we have that $Z_\alpha u_1$ is a subsequence of
$$X_1[i_1] := x[1, 1]x[1, 2] \ldots x[1, i_1],$$
$$X_2[i_2] := x[2, 1]x[2, 2] \ldots x[2, i_2],$$
$$\ldots\ldots\ldots\ldots$$
$$X_s[i_s] := x[s, 1]x[s, 2] \ldots x[s, i_s],$$
and a suffix of
$$Y_1[j_1] := y[1, 1]y[1, 2] \ldots y[1, j_1],$$
$$Y_2[j_2] := y[2, 1]y[2, 2] \ldots y[2, j_2],$$
$$\ldots\ldots\ldots\ldots$$
$$Y_t[j_t] := y[t, 1]y[t, 2] \ldots y[t, j_t].$$
By the definition of $Z[i_1, i_2, \ldots, i_s; j_1, j_2, \ldots, j_t]$, we have that
$$2 \le |Z[i_1 - 1, i_2 - 1, \ldots, i_s - 1; j_1 - 1, j_2 - 1, \ldots, j_t - 1]| + 1$$
$$= |Z_\alpha u_1| = |Z_\alpha| + 1 \le |Z[i_1, i_2, \ldots, i_s; j_1, j_2, \ldots, j_t]|.$$
By the definition of $Z_\beta := Z[i_1, i_2, \ldots, is; j_1, j_2, \ldots, j_t]$, we have that the last element, say u_2, in Z_β must be equal to $y[1, j_1]$, $y[2, j_2]$, ..., and $y[t, j_t]$. Thus $u_1 = u_2 = x[1, i_1] = x[2, i_2] = \cdots = x[s, i_s]$. Therefore $Z_\beta - u_2$, which is a string obtained from removing u_2 from Z_β, is a subsequence of
$$X_1[i_1 - 1] := x[1, 1]x[1, 2] \ldots x[1, i_1 - 1],$$
$$X_2[i_2 - 1] := x[2, 1]x[2, 2] \ldots x[2, i_2 - 1],$$
$$\ldots\ldots\ldots\ldots$$
$$X_s[i_s - 1] := x[s, 1]x[s, 2] \ldots x[s, i_s - 1],$$
and a suffix of
$$Y_1[j_1 - 1] := y[1, 1]y[1, 2] \ldots y[1, j_1 - 1],$$
$$Y_2[j_2 - 1] := y[2, 1]y[2, 2] \ldots y[2, j_2 - 1],$$
$$\ldots\ldots\ldots\ldots$$
$$Y_t[j_t - 1] := y[t, 1]y[t, 2] \ldots y[t, j_t - 1].$$
By the definition of $Z[i_1 - 1, i_2 - 1, \ldots, i_s - 1; j_1 - 1, j_2 - 1, \ldots, j_t - 1]$, we have that
$$|Z[i_1, i_2, \ldots, i_s; j_1, j_2, \ldots, j_t]| - 1 = |Z_\beta - u_2|$$
$$= |Z_\beta| - 1 \le |Z[i_1 - 1, i_2 - 1, \ldots, i_s - 1; j_1 - 1, j_2 - 1, \ldots, j_t - 1]|.$$
Therefore
$$|Z[i_1, i_2, \ldots, i_s; j_1, j_2, \ldots, j_t]| = |Z[i_1 - 1, i_2 - 1, \ldots, i_s - 1; j_1 - 1, j_2 - 1, \ldots, j_t - 1]| + 1.$$
Case 2. $Z[i_1 - 1, i_2 - 1, \ldots, i_s - 1; j_1 - 1, j_2 - 1, \ldots, j_t - 1]$ is empty.
Since u_1 is a subsequence of
$$X_1[i_1] := x[1, 1]x[1, 2] \ldots x[1, i_1],$$
$$X_2[i_2] := x[2, 1]x[2, 2] \ldots x[2, i_2],$$
$$\ldots\ldots\ldots\ldots$$
$$X_s[i_s] := x[s, 1]x[s, 2] \ldots x[s, i_s],$$

and a suffix of

$$Y_1[j_1] := y[1, 1]y[1, 2] \ldots y[1, j_1],$$
$$Y_2[j_2] := y[2, 1]y[2, 2] \ldots y[2, j_2],$$
$$\ldots\ldots\ldots\ldots$$
$$Y_t[j_t] := y[t, 1]y[t, 2] \ldots y[t, j_t].$$

By the definition of $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$, we have that

$$1 = |u_1| \leq |Z[i_1, i_2, ..., i_s; j_1, j_2, ..., jt]|.$$

Notice that the proofs for

$$Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| - 1 \leq |Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1 - 1, j_2 - 1, ..., j_t - 1]|$$

in the above Case 1 still hold in this case. We have

$$0 \leq |Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| - 1 \leq |Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1 - 1, j_2 - 1, ..., j_t - 1]| = 0.$$

Thus

$$|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| = 1,$$
$$|Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1 - 1, j_2 - 1, ..., j_t - 1]| = 0.$$

Therefore

$$|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| = |Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1 - 1, j_2 - 1, ..., j_t - 1]| + 1.$$

Hence the proof of Claim 2 is complete.

Claim 3. Suppose that $v_1 := y[1, j_1] = y[2, j_2] = \cdots = y[t, j_t]$. If $v_1 \neq x[1, i_1]$, $v_1 \neq x[2, i_2]$, ..., and $v_1 \neq x[s, i_s]$, then

$$|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| = |Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1, j_2, ..., j_t]|.$$

Proof of Claim 3. Our proof is divided into two cases again.

Case 1. $Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1, j_2, ..., j_t]$ is not empty.

Notice that $Z_\gamma := Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1, j_2, ..., j_t]$ is a subsequence of

$$X_1[i_1 - 1] := x[1, 1]x[1, 2] \ldots x[1, i_1 - 1],$$
$$X_2[i_2 - 1] := x[2, 1]x[2, 2] \ldots x[2, i_2 - 1],$$
$$\ldots\ldots\ldots\ldots$$
$$X_s[i_s - 1] := x[s, 1]x[s, 2] \ldots x[s, i_s - 1],$$

and a suffix of

$$Y_1[j_1] := y[1, 1]y[1, 2] \ldots y[1, j_1],$$
$$Y_2[j_2] := y[2, 1]y[2, 2] \ldots y[2, j_2],$$
$$\ldots\ldots\ldots\ldots$$
$$Y_t[j_t] := y[t, 1]y[t, 2] \ldots y[t, j_t].$$

Then Z_γ is a subsequence of

$$X_1[i_1] := x[1, 1]x[1, 2] \ldots x[1, i_1],$$
$$X_2[i_2] := x[2, 1]x[2, 2] \ldots x[2, i_2],$$
$$\ldots\ldots\ldots\ldots$$
$$X_s[i_s] := x[s, 1]x[s, 2] \ldots x[s, i_s],$$

and a suffix of

$$Y_1[j_1] := y[1, 1]y[1, 2] \ldots y[1, j_1],$$
$$Y_2[j_2] := y[2, 1]y[2, 2] \ldots y[2, j_2],$$
$$\ldots\ldots\ldots\ldots$$
$$Y_t[j_t] := y[t, 1]y[t, 2] \ldots y[t, j_t].$$

By the definition of $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$, we have that

$$|Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1, j_2, ..., j_t]| = |Z_\gamma| \leq |Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]|.$$

Since $Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1, j_2, ..., j_t]$ is not empty, $Z_\delta := Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ is not empty. Thus the last element, say v_2, in Z_δ must be equal to $y[1, j_1], y[2, j_2], ..., y[t, j_t]$. Thus $v_1 = v_2 = y[1, j_1] = y[2, j_2] = \cdots = y[t, j_t]$. Since $v_2 = v_1 \neq x[1, i_1]$, $v_2 = v_1 \neq x[2, i_2]$, ..., $v_2 = v_1 \neq x[s, i_s]$, we have that Z_δ is a subsequence of

$$X_1[i_1 - 1] := x[1, 1]x[1, 2] \ldots x[1, i_1 - 1],$$
$$X_2[i_2 - 1] := x[2, 1]x[2, 2] \ldots x[2, i_2 - 1],$$
$$\ldots\ldots\ldots\ldots$$
$$X_s[i_s - 1] := x[s, 1]x[s, 2] \ldots x[s, i_s - 1],$$

and a suffix of

$$Y_1[j_1] := y[1, 1]y[1, 2] \ldots y[1, j_1],$$
$$Y_2[j_2] := y[2, 1]y[2, 2] \ldots y[2, j_2],$$
$$\ldots\ldots\ldots\ldots$$
$$Yt[j_t] := y[t, 1]y[t, 2] \ldots y[t, j_t].$$

By the definition of $Z[i_1 - 1, i_2 - 2, ..., i_s - 1; j_1, j_2, ..., j_t]$, we have that

$$|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| \leq |Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1, j_2, ..., j_t]|.$$

Therefore

$$|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| = |Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1, j_2, ..., j_t]|.$$

Case 2. $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ is empty.

Our assertion is that $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ must be empty. Suppose, to the contrary, that $Z_v := Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ is not empty. Then the last element, say v_3, in Z_v must be equal to $y[1, j_1], y[2, j_2], ..., y[t, j_t]$. Thus $v_1 = v_3 = y[1, j_1] = y[2, j_2] = \cdots = y[t, j_t]$. Since $v_3 = v_1 \neq x[1, i_1]$, $v_3 = v_1 \neq x[2, i_2]$, ..., $v_3 = v_1 \neq x[s, i_s]$, we have that Z_v is a subsequence of

$$X_1[i_1 - 1] := x[1, 1]x[1, 2] \ldots x[1, i_1 - 1],$$
$$X_2[i_2 - 1] := x[2, 1]x[2, 2] \ldots x[2, i_2 - 1],$$
$$\ldots\ldots\ldots\ldots$$
$$X\_{}_s[i_s - 1] := x[s, 1]x[s, 2] \ldots x[s, i_s - 1],$$

and a suffix of

$$Y_1[j_1] := y[1, 1]y[1, 2] \ldots y[1, j_1],$$
$$Y_2[j2] := y[2, 1]y[2, 2] \ldots y[2, j_2],$$
$$\ldots\ldots\ldots\ldots$$
$$Y_t[j_t] := y[t, 1]y[t, 2] \ldots y[t, j_t].$$

By the definition of $Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1, j_2, ..., j_t]$, we have

$$1 \leq |Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| \leq |Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1, j_2, ..., j_t]| = 0,$$

a contradiction. Thus $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ is empty and

$$|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| = |Z[i_1 - 1, i_2 - 1, ..., i_s - 1; j_1, j_2, ..., j_t]| = 0.$$

Hence the proof of Claim 3 is complete.

Claim 4. Suppose that $w_1 := y[1, j_1] = y[2, j_2] = \cdots = y[t, j_t]$. Assume that w_1 is not equal to exactly r elements in the set $L: = \{x[1, i_1], x[2, i_2], ..., x[s, i_s]\}$, where $1 \leq r \leq (s - 1)$. Without loss of generality, we assume w_1 is not equal to exactly the first r elements in L. Namely, $w_1 \neq x[1, i_1]$, $w_1 \neq x[2, i_2]$, ..., $w_1 \neq x[r, i_r]$, $w_1 = x[r + 1, i_{r+1}] = x[r + 2, i_{r+2}] = \cdots = x[s, i_s]$. Then

$$|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| = |Z[i_1 - 1, i_2 - 1, ..., i_r - 1, i_{r+1}, i_{r+2}, ..., i_s; j_1, j_2, ..., j_t]|.$$

Proof of Claim 4. Our proof is divided into two cases.

Case 1. $Z[i_1 - 1, i_2 - 1, ..., i_r - 1, i_{r+1}, i_{r+2}, ..., i_s; j_1, j_2, ..., j_t]$ is not empty.
Notice that $Z_\varepsilon := Z[i_1 - 1, i_2 - 1, ..., i_r - 1, i_{r+1}, i_{r+2}, ..., i_s; j_1, j_2, ..., j_t]$ is a subsequence of

$$X_1[i_1 - 1] := x[1, 1]x[1, 2] ... x[1, i_1 - 1],$$
$$X_2[i_2 - 1] := x[2, 1]x[2, 2] ... x[2, i_2 - 1],$$
$$............$$
$$X_r[i_r - 1] := x[r, 1]x[r, 2] ... x[r, i_r - 1],$$
$$X_{r+1}[i_{r+1}] := x[r + 1, 1]x[r + 1, 2] ... x[r + 1, i_{r+1}],$$
$$X_{r+2}[i_{r+2}] := x[r + 2, 1]x[r + 2, 2] ... x[r + 2, i_{r+2}],$$
$$............$$
$$X_s[i_s] := x[s,1]x[s, 2] ... x[s, i_s],$$

and a suffix of

$$Y_1[j_1] := y[1, 1]y[1, 2] ... y[1, j_1],$$
$$Y_2[j_2] := y[2, 1]y[2, 2] ... y[2, j_2],$$
$$............$$
$$Y_t[j_t] := y[t, 1]y[t, 2] ... y[t, j_t].$$

Then Z_ε is a subsequence of

$$X_1[i_1] := x[1, 1]x[1, 2] ... x[1, i_1],$$
$$X_2[i_2] := x[2, 1]x[2, 2] ... x[2, i_2],$$
$$............$$
$$X_s[i_s] := x[s, 1]x[s, 2] ... x[s, i_s],$$

and a suffix of

$$Y_1[j_1] := y[1, 1]y[1, 2] ... y[1, j_1],$$
$$Y_2[j_2] := y[2, 1]y[2, 2] ... y[2, j_2],$$
$$............$$
$$Y_t[j_t] := y[t, 1]y[t, 2] ... y[t, j_t].$$

By the definition of $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$, we have that

$$|Z[i_1 - 1, i_2 - 1, ..., i_r - 1, i_{r+1}, i_{r+2}, ..., i_s; j_1, j_2, ..., j_t]| = |Z_\varepsilon| \leq |Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]|.$$

Since $Z[i_1 - 1, i_2 - 1, ..., i_r - 1, i_{r+1}, i_{r+2}, ..., i_s; j_1, j_2, ..., j_t]$ is not empty, $Z_\mu := Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ is not empty. Thus the last element, say w_2, in Z_μ must be equal to $y[1, j_1]$, $y[2, j_2]$, ..., and $y[t, j_t]$. Thus $w_1 = w_2 \neq x[1, i_1]$, $w_1 = w_2 \neq x[2, i_2]$, ..., $w_1 = w_2 \neq x[r, i_r]$, and $w_1 = w_2 = x[r + 1, i_{r+1}] = x[r + 2, i_{r+2}] = \cdots = x[s, i_s]$. Therefore Z_μ is a subsequence of

$$X_1[i_1 - 1] := x[1, 1]x[1, 2] ... x[1, i_1 - 1],$$
$$X_2[i_2 - 1] := x[2, 1]x[2, 2] ... x[2, i_2 - 1],$$
$$............$$
$$X_r[i_r - 1] := x[r, 1]x[r, 2] ... x[r, i_r - 1],$$
$$X_{r+1}[i_{r+1}] := x[r + 1, 1]x[r + 1, 2] ... x[r + 1, i_{r+1}],$$
$$X_{r+2}[i_{r+2}] := x[r + 2, 1]x[r + 2, 2] ... x[r + 2, i_{r+2}],$$
$$............$$
$$X_s[i_s] := x[s, 1]x[s, 2] ... x[s, i_s],$$

and a suffix of

$$Y_1[j_1] := y[1, 1]y[1, 2] ... y[1, j_1],$$
$$Y_2[j_2] := y[2, 1]y[2, 2] ... y[2, j_2],$$
$$............$$
$$Y_t[j_t] := y[t, 1]y[t, 2] ... y[t, j_t].$$

By the definition of $Z[i_1 - 1, i_2 - 1, ..., i_r - 1, i_{r+1}, i_{r+2}, ..., i_s; j_1, j_2, ..., j_t]$, we have that

$$|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| \leq |Z[i_1 - 1, i_2 - 1, ..., i_r - 1, i_{r+1}, i_{r+2}, ..., i_s; j_1, j_2, ..., j_t]|.$$

Therefore

$$|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| = |Z[i_1 - 1, i_2 - 1, ..., i_r - 1, i_{r+1}, i_{r+2}, ..., i_s; j_1, j_2, ..., j_t]|.$$

Case 2. $Z[i_1 - 1, i_2 - 1, ..., i_r - 1, i_{r+1}, i_{r+2}, ..., i_s; j_1, j_2, ..., j_t]$ is empty.

Our assertion is that $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ must be empty. Suppose, to the contrary, that $Z_\rho := Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ is not empty. Then the last element, say w_3, in Z_ρ must be equal to $y[1, j_1], y[2, j_2], ..., y[t, j_t]$. Thus $w_1 = w_3 \neq x[1, i_1]$, $w_1 = w_3 \neq x[2, i_2]$, ..., $w_1 = w_3 \neq x[r, i_r]$, and $w_1 = w_3 = x[r + 1, i_{r+1}] = x[r + 2, i_{r+2}] = \cdots = x[s, i_s]$. Therefore Z_ρ is a subsequence of

$$X_1[i_1 - 1] := x[1, 1]x[1, 2] ... x[1, i_1 - 1],$$
$$X_2[i_2 - 1] := x[2, 1]x[2, 2] ... x[2, i_2 - 1],$$
$$...........$$
$$X_r[i_r - 1] := x[r, 1]x[r, 2] ... x[r, i_r - 1],$$
$$X_{r+1}[i_{r+1}] := x[r + 1, 1]x[r + 1, 2] ... x[r + 1, i_{r+1}],$$
$$X_{r+2}[i_{r+2}] := x[r + 2, 1]x[r + 2, 2] ... x[r + 2, i_{r+2}],$$
$$...........$$
$$X_s[i_s] := x[s, 1]x[s, 2] ... x[s, i_s],$$

and a suffix of

$$Y_1[j_1] := y[1, 1]y[1, 2] ... y[1, j_1],$$
$$Y_2[j_2] := y[2, 1]y[2, 2] ... y[2, j_2],$$
$$...........$$
$$Y_t[j_t] := y[t, 1]y[t, 2] ... y[t, j_t].$$

By the definition of $Z[i_1 - 1, i_2 - 1, ..., i_r - 1, i_{r+1}, i_{r+2}, ..., i_s; j_1, j_2, ..., j_t]$, we have that

$$1 \leq |Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| \leq |Z[i_1 - 1, i_2 - 1, ..., i_r - 1, i_{r+1}, i_{r+2}, ..., i_s; j_1, j_2, ..., j_t]| = 0,$$

a contradiction. Thus $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ is empty and

$$| Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]|$$
$$= | Z[i_1 - 1, i_2 - 1, ..., i_r - 1, i_{r+1}, i_{r+2}, ..., i_s; j_1, j_2, ..., j_t]| = 0.$$

Hence the proof of Claim 4 is complete.

Remark 1. The general form of Claim 4 is as follows.

Claim 4'. Suppose that $w_1 := Y[1, j_1] = Y[2, j_2] =, ..., = Y[t, j_t]$. If $w_1 \neq x[\pi(1), i_{\pi(1)}]$, $w_1 \neq x[\pi(2), i_{\pi(2)}]$, ..., $w_1 \neq x[\pi(r), i_{\pi(r)}]$, where $\pi(1), \pi(2), ..., \pi(r)$ are integers such that $1 \leq \pi(1) < \pi(2) < \cdots < \pi(r) \leq s$, and for any $e \in \{1, 2, ..., s\} - \{\pi(1), \pi(2), ..., \pi(r)\}$, $w_1 = x[e, i_e]$, then

$$|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]|$$
$$= |Z[i_1, ..., i_{\pi(1)-1}, i_{\pi(1)} - 1, i_{\pi(1)+1}, ..., i_{\pi(2)-1}, i_{\pi(2)} - 1, i_{\pi(2)+1}, ..., i_{\pi(r)-1}, i_{\pi(r)} - 1, i_{\pi(r)+1}, ..., i_s; j_1, j_2, ..., j_t]|.$$

Remark 2. Suppose that $w_1 := y[1, j_1] = y[2, j_2] = \cdots = y[t, j_t]$. We need to follow Claim 2, Claim 3, and Claim 4 to compute $|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]|$. The largest number of formats we can encounter is

$$C(s, 0) + C(s, 1) + C(s, 2) + \cdots + C(s, s) = 2^s,$$

where $C(s, a)$ denotes the number of a-element subsets of a set of size s, where a is an integer such that $0 \leq a \leq s$.

Claim 5. Let $H = h_1 h_2 ... h_b$ be a longest string which is a subsequence of $X_1, X_2, ..., X_s$, and a substring of $Y_1, Y_2, ..., Y_t$. Then

$$b = \max\{|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| : 1 \le i_1 \le m_1, 1 \le i_2 \le m_2, ..., 1 \le i_s \le m_s, 1 \le j_1 \le n_1, 1 \le j_2 \le n_2, ..., 1 \le j_t \le n_t\},$$

where $m_u = |X_u|$ for each u with $1 \le u \le s$ and $n_v = |Y_v|$ for each v with $1 \le v \le t$.

Proof of Claim 5. For any $i_1, i_2, ..., i_s$ with $1 \le i_1 \le m_1, 1 \le i_2 \le m_2, ..., 1 \le i_s \le m_s$, and any $j_1, j_2, ..., j_t$ with $1 \le j_1 \le n_1, 1 \le j_2 \le n_2, ..., 1 \le j_t \le n_t$, we, from the definition of $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j\_t]$, have that $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$ is a subsequence of $X_1, X_2, ..., X_s$, and a substring of $Y_1, Y_2, ..., Y_n$. By the definition of H, we have that

$$|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| \le |H| = b.$$

Thus

$$\max\{|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| : 1 \le i_1 \le m_1, 1 \le i_2 \le m_2, ..., 1 \le i_s \le m_s, 1 \le j_1 \le n_1, 1 \le j_2 \le n_2, ..., 1 \le j_t \le n_t\} \le b.$$

Since $H = h_1 h_2 ... h_b$ is a longest string which is a subsequence of $X_1, X_2, ..., X_s$, and a substring of $Y_1, Y_2, ..., Y_n$, there exits indices $i_1, i_2, ..., i_s$ and indices $j_1, j_2, ..., j_t$ such that $h_b = x[1, i_1], h_b = x[2, i_2], ..., h_b = x[s, i_s]$, and $h_b = y[1, j_1], h_b = y[2, j_2], ..., h_b = y[t, j_t]$. Thus $H = h_1 h_2 ... h_b$ is a subsequence of $X_1[i_1], X_2[i_2], ..., X_s[i_s]$ and a suffix of $Y_1[j_1], Y_2[j_2], ..., Y_t[j_t]$. From the definition of $Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]$, we have that

$$b \le |Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| \le \max \{|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| : 1 \le i_1 \le m_1, 1 \le i_2 \le m_2, ..., 1 \le i_s \le m_s, 1 \le j_1 \le n_1, 1 \le j_2 \le n_2, ..., 1 \le j_t \le n_t\}.$$

Therefore

$$b = \max \{|Z[i_1, i_2, ..., i_s; j_1, j_2, ..., j_t]| : 1 \le i_1 \le m_1, 1 \le i_2 \le m_2, ..., 1 \le i_s \le m_s, 1 \le j_1 \le n_1, 1 \le j_2 \le n_2, ..., 1 \le j_t \le n_t\}.$$

Hence the proof of Claim 5 is complete.

3. An algorithm on the (s, t)-longest common subsequence and substring problem

Based on Claims 1-5 in Section 2, we can design an algorithm for the (s, t)-longest common subsequence and substring problem. Once again, we assume that $X_p = x[p, 1]x[p, 2] ... x[p, m_p]$, where $x[p, a]$ with p is an integer such that $1 \le p \le s$ and $1 \le a \le m_p$ are elements in the alphabet $\sum$, are s strings, and $Y_q = y[q, 1]y[q, 2] ... y[q, n_q]$, where $y[q, b]$ with q is an integer such that $1 \le q \le t$ and $1 \le b \le n_q$ are elements in the alphabet $\sum$, are t strings. In the following Algorithm A, W is an $(m_1 + 1)(m_2 + 1)\cdots(m_s + 1)(n_1 + 1)(n_2 + 1) \dots (n_t + 1)$ dimensional array and the cells $W(i_1, i_2, ..., i_s, j_1, j_2, ..., j_t)$, where $1 \le i_1 \le m_1, 1 \le i_2 \le m_2, ..., 1 \le i_s \le m_s$, and $1 \le j_1 \le n_1, 1 \le j_2 \le n_2, ..., 1 \le j_t \le n_t$ store the lengths of strings such that each of them satisfies the following conditions.

(1.1) It is a subsequence of $X_1[i_1] = x[1, 1]x[1, 2] ... x[1, i_1]$.
(1.2) It is a subsequence of $X_2[i_2] = x[2, 1]x[2, 2] ... x[2, \_{2}]$.
....
(1.s) It is a subsequence of $X_s[i_s] = x[s, 1]x[s, 2] ... x[s, i_s]$.
(2.1) It is a suffix of $Y_1[j_1] = y[1, 1]y[1, 2] ... y[1, j_1]$.
(2.2) It is a suffix of $Y_2[j_2] = y[2, 1]y[2, 2] ... y[2, j_2]$.
....
(2.t) It is a suffix of $Y_t[j_t] = y[t, 1]y[t, 2] ... y[t, j_t]$.

(3.1) Under the conditions above, its length is as large as possible.

ALG A($X_1, X_2, ..., X_m, Y_1, Y_2, ..., Y_n,$ m, n, W)

1. Initialization:

$W(i_1, i_2, ..., i_s, j_1, j_2, ..., j_t) \leftarrow 0$, where $0 \le i_1 \le m_1$, $i_2 = 0$, $i_3 = 0$, ..., $i_s = 0$, $j_1 = 0$, $j_2 = 0$, $j_3 = 0$, ..., $j_t = 0$.

$W(i_1, i_2, ..., i_s, j_1, j_2, ..., j_t) \leftarrow 0$, where $i_1 = 0$, $0 \le i_2 \le m_2$, $i_3 = 0$, ..., $i_s = 0$, $j_1 = 0$, $j_2 = 0$, $j_3 = 0$, ..., $j_t = 0$.

....

$W(i_1, i_2, ..., i_s, j_1, j_2, ..., j_t) \leftarrow 0$, where $i_1 = 0$, $i_2 = 0$, ..., $i_{s-1} = 0$, $0 \le i_s \le m_s$, $j_1 = 0$, $j_2 = 0$, $j_3 = 0$, ..., $j_t = 0$.

$W(i_1, i_2, ..., i_s, j_1, j_2, ..., j_t) \leftarrow 0$, where $i_1 = 0$, $i_2 = 0$, $i_3 = 0$, $i_4 = 0$, ..., $i_s = 0$, $0 \le j_1 \le n_1$, $j_2 = 0$, $j_3 = 0$, ..., $j_t = 0$.

$W(i_1, i_2, ..., i_s, j_1, j_2, ..., j_t) \leftarrow 0$, where $i_1 = 0$, $i_2 = 0$, $i_3 = 0$, $i_4 = 0$, ..., $i_s = 0$, $j_1 = 0$, $0 \le j_2 \le n_2$, $j_3 = 0$, ..., $j_t = 0$.

....

$W(i_1, i_2, ..., i_s, j_1, j_2, ..., j_t) \leftarrow 0$, where $i_1 = 0$, $i_2 = 0$, $i_3 = 0$, $i_4 = 0$, ..., $i_s = 0$, $j_1 = 0$, $j_2 = 0$, $j_3 = 0$, ..., $0 \le j_t \le n_t$.

maxLength = 0.

lastIndexOnY1 = n_1.

2.1. **for** $\theta_1 \leftarrow 1$ **to** m_1

2.2. **for** $\theta_2 \leftarrow 1$ **to** m_2

....

2.s. **for** $\theta_s \leftarrow 1$ **to** m_s

3.1. **for** $\tau_1 \leftarrow 1$ **to** n_1

3.2. **for** $\tau_2 \leftarrow 1$ **to** n_2

....

3.t. **for** $\tau_t \leftarrow 1$ **to** n_t

if $y[1, \tau_1], y[2, \tau_2], ..., y[t, \tau_t]$ are not the same

$W(\theta_1, \theta_2, ..., \theta_s, \tau_1, \tau_2, ..., \tau_t) \leftarrow 0$

else

Set $\sigma := y[1, \tau_1] = y[2, \tau_2] = x[t, \tau_t]$

if $\sigma = x[1, \theta_1] = x[2, \theta_2] = \cdots = x[s, \theta_s]$

$W(\theta_1, \theta_2, ..., \theta_s, \tau_1, \tau_2, ..., \tau_t) \leftarrow$
$W(\theta_1 - 1, \theta_2 - 1 ..., \theta_s - 1, \tau_1 - 1, \tau_2 - 1 ..., \tau_t - 1) + 1$

else if $\sigma \ne x[1, \theta_1]$, $\sigma \ne x[2, \theta_2]$, ..., $\sigma \ne x[s, \theta_s]$,

$W(\theta_1, \theta_2, ..., \theta_s, \tau_1, \tau_2, ..., \tau_t) \leftarrow$
$W(\theta_1 - 1, \theta_2 - 1 ..., \theta_s - 1, \tau_1, \tau_2, ..., \tau_t)$

else $\sigma \ne x[\pi(1), i_{\pi(1)}]$, $\sigma \ne x[\pi(2), i_{\pi(2)}]$, ..., $\sigma \ne x[\pi(r), i_{\pi(r)}]$, where $1 \le \pi(1) < \pi(2) < \cdots < \pi(r) \le s$, $1 \le r \le (s - 1)$, and for any $e \in \{1, 2, ..., s\} - \{\pi(1), \pi(2), ..., \pi(r)\}$, $\sigma = x[e, \theta_e]$,

$W(\theta_1, \theta_2, ..., \theta_s, \tau_1, \tau_2, ..., \tau_t) \leftarrow$
$W(\theta_1, ..., \theta_{\pi(1)-1}, \theta_{\pi(1)} - 1, \theta_{\pi(1)+1}, ..., \theta_{\pi(2)-1}, \theta_{\pi(2)} - 1, \theta_{\pi(2)+1}, ..., \theta_{\pi(r)-1}, \theta_{\pi(r)} - 1, \theta_{\pi(r)+1}, ..., \theta s; \tau_1, \tau_2, ..., \tau_t)$

if $W(\theta_1, \theta_2, ..., \theta_s, \tau_1, \tau_2, ..., \tau_t) >$ maxLength
maxLength = $W(\theta_1, \theta_2, ..., \theta_s, \tau_1, \tau_2, ..., \tau_t)$
lastIndexOnY1 = τ_1

4. **return** A substring Y_1 between (lastIndexOnY1 - maxLength) and lastIndexOnY1.

Because of Claims 1-5 in Section 2, we have that Algorithm A is correct. We also have the following result on the time and space complexities of Algorithm A.

Theorem 1. Both the time complexity and the space complexity of Algorithm A are

$$O(m_1 m_2 \cdots m_s n_1 n_2 \cdots n_t) = O(M^{s+t}),$$

where $M = \max\{m_1, m_2, ..., m_s, n_1, n_2, ..., n_t\}$.

4. Conclusion

In this paper, we introduce the longest common subsequence and substring problem for multiple strings. We propose an algorithm to solve the problem. In the future, we will design new algorithms to improve the time and space complexities of our algorithm and find the applications of our algorithm in the real world.

Acknowledgements. The author would like to thank the referee for his or her suggestions leading to the improvements of the initial manuscript.

Author's Contributions: This work represents the sole contribution of the author.

Conflicts of interest. The author declares no conflicts of interest.

REFERENCES

1. A. Apostolico, String editing and longest common subsequences, in: G. Rozenberg and A. Salomaa (Eds.), Linear Modeling: Background and Application, in: Handbook of Formal Languages, Vol. 2, Springer-Verlag, Berlin, 1997.
2. A. Apostolico, Chapter 13: General pattern matching, in: M. J. Atallah (Ed.), Handbook of Algorithms and Theory of Computation, CRC, Boca Raton, FL, 1998.
3. L. Bergroth, H. Hakonen, and T. Raita, A survey of longest common subsequence algorithms, in: SPIRE, A Corua, Spain, 2000.
4. C. Blum, M. Djukanovic, A. Santini, H. Jiang, C. Li, F. Manyà, and G. R. Raidl, Solving longest common subsequence problems via a transformation to the maximum clique problem, *Computers and Operations Research*, 125 (2021) 105089.
5. T. Cormen, C. Leiserson, and R. Rivest, Section 16.3: Longest common subsequence, Introduction to Algorithms, MIT Press, Cambridge, MA, 1990.
6. M. Crochemore, C. S. Iliopoulos, A. Langiu, and F. Mignosi, The longest common substring problem, *Mathematical Structures in Computer Science*, pp 1-19, Cambridge University Press 2015, doi:10.1017/S0960129515000110.
7. D. Gusfield, II: Suffix Trees and Their Uses, Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology, Cambridge University Press, 1997.
8. D. Hirschberg, A linear space algorithm for computing maximal common subsequences, *Communications of the ACM*, 18 (1975) 341-343.

9. D. Hirschberg, Serial computations of Levenshtein distances, in: A. Apostolico and Z. Galil (Eds.), Pattern Matching Algorithms, Oxford University Press, Oxford, 1997.
10. J. Hunt and T. Szymanski, A fast algorithm for computing longest common subsequences, *Communications of the ACM*, 20 (1977) 350-353.
11. R. Li, J. Deka, and K. Deka, An algorithm for the longest common subsequence and substring problem, *Journal of Mathematics and Informatics*, 25 (2023) 77-81.
12. S. R. Mousavi and F. Tabataba, An improved algorithm for the longest common subsequence problem, *Computers and Operations Research*, 39 (2012) 512-520.
13. C. Rick, New algorithms for the longest common subsequence problem, Research Report No. 85123-CS, University of Bonn, 1994.
14. P. Weiner, Linear pattern matching algorithms. In: 14th Annual Symposium on Switching and Automata Theory, Iowa City, Iowa, USA, October 15–17, 1973, pp. 1–11 (1973).