Longest (Sub-)Periodic Subsequence

Hideo Bannai      Tomohiro I      Dominik Köppl

Abstract
We present an algorithm computing the longest periodic subsequence of a string of length $n$ in $O(n^7)$ time with $O(n^4)$ words of space. We obtain improvements when restricting the exponents or extending the search allowing the reported subsequence to be subperiodic down to $O(n^3)$ time and $O(n^2)$ words of space.

1 Introduction
A natural extension of the analysis of regularities such as squares or palindromes perceived as substrings of a given text is the study of the same type of regularities when considering subsequences. In this line of research, given a text of length $n$, Kosowski [17] proposed an algorithm running in $O(n^2)$ time using $O(n)$ words of space to find the longest subsequence that is a square. Inoue et al. [13] generalized this setting to consider the longest such subsequence common of two texts $T$ and $S$ of length $n$, and gave an algorithm computing this sequence in $O(n^6)$ time using $O(n^4)$ space, also providing improvements in case that the number of matching characters pairs $T[i] = S[j]$ is rather small. Recently, Inoue et al. [14] provided similar improvements for the longest square subsequence of a single string. Here, we consider the problem for a single text, but allow the subsequence to have different exponents. In detail, we want to find the longest subsequence that is (sub-)periodic.

A non-exhaustive list of related problems are finding the longest palindromic subsequence [5, 12], absent subsequences [16], longest increasing and decreasing subsequences [7, 19], maximal common subsequences [6, 18], the longest run subsequence [20], the longest Lyndon subsequence [2], longest rollercoasters [3, 8, 9], and computing subsequence automaton [11, 4]. Our techniques rely on finding longest common subsequences, which is conceived as a well-studied problem (see [10, 11, 15] and the references therein).

2 Preliminaries
Let $\mathbb{N}$ denote all natural numbers $1, 2, \ldots$, and $\mathbb{Q}^+$ the set of all rational numbers greater than or equal to 1. We distinguish integer intervals $1, \ldots, n = [1..n]$ and intervals of rational numbers $[1/2, 3/4]$. 

1
Table 1: Space and time complexities for finding periodic subsequences of specific exponents. $\epsilon$ is a rational number with $0 < \epsilon < 1$. The exponent column means that a subsequence is considered only if it has at least one exponent within the domain given in the column.

| exponent          | time   | space   | solution |
|--------------------|--------|---------|----------|
| $2\mathbb{N}$      | $O(n^2)$ | $O(n)$  | Thm. 3.1 |
| $\mathbb{N} + \epsilon$ | $O(n^3)$ | $O(n^2)$ | Thm. 3.2 |
| $\mathbb{Q}^+ \cap ((2, 3] \cup [4, \infty))$ | $O(n^3)$ | $O(n^2)$ | Thm. 4.2 |
| $(3 + \epsilon)\mathbb{N}$ | $O(n^7)$ | $O(n^4)$ | Thm. 4.3 |

Let $\Sigma$ denote a totally ordered set of symbols called the alphabet. An element of $\Sigma^*$ is called a string. Given a string $S \in \Sigma^*$, we denote its length with $|S|$ and its $i$-th symbol with $S[i]$ for $i \in [1..|S|]$. Further, we write $S[i..j] = S[i] \cdots S[j]$. A factorization $T = F_1 \cdots F_k$ is a partitioning of $T$ into substrings $F_1, \ldots, F_k$. A subsequence of a string $S$ with length $\ell$ is a string $S[i_1] \cdots S[i_\ell]$ with $i_1 < \ldots < i_\ell$.

Given a string $S$, we can write $S$ in the form $S = U^x U'$ with $U'$ being either empty or a proper prefix of $U$. Then $|U|$ is called a period of $S$, and $x + |U'|/|U| \in \mathbb{Q}^+$ is called its exponent with respect to the period $|U|$. For the largest possible such exponent, $S$ is called periodic if $x \geq 2$, or sub-periodic if $x \in (1, 2)$. For instance, the unary string $T = a \cdots a$ has the minimum period 1 with exponent $|T|$, or more generally, period $p \in [1..|T|]$ with exponent $|T|/p$.

Further, for two strings $Y$ and $Z$, let $\text{LCS}_{Y,Z}(y, z)$ denote the longest common subsequence of $Y[1..y]$ and $Z[1..z]$. We omit the strings in subscript if they are clear from the context. Also, we allow us to write LCS for an arbitrary number of strings; the number of strings is reflected by the arguments given to LCS. It is known that we can answer the longest common subsequence $\text{LCS}_{X_1, \ldots, X_k}(x_1, \ldots, x_k)$ of $k$ strings $X_1[1..x_1], \ldots, X_k[1..x_k]$ in constant time by building a table of size $O(n^k)$ and filling its entries in $O(kn^k)$ time via dynamic programming.

**Structure of the Paper** In what follows, we first show an algorithm (Sect. 3) computing the longest subsequence that is periodic or sub-periodic. Subsequently, we refine this algorithm to omit the sub-periodic subsequences by allowing more time and space in Sect. 4. Table 1 gives an overview of our obtained results. We observe that, if we are only interested in the longest subsequence having an exponent $> 1$, we obtain a algorithm faster than those finding subsequences with more restricted exponents.

A key observation is that a longest (sub-)periodic subsequence $S$ is maximal, meaning that no occurrence $T[i_1]T[i_2] \cdots T[i_{|S|}]$ of $S$ in $T$ can be extended with a character in $T[1..i_1 - 1]$ or $T[i_{|S|} + 1..]$ to form a longer subsequence without breaking the periodicity.
Figure 1: $D_{2}$ of Sect. 3 for the text $T = \text{abcaaaca}$ with the factorization $T = YZ$ with $Y = \text{abcaa}$ and $Z = \text{aca}$.

3 Longest (Sub-)periodic Subsequence

We start with the search for the longest subsequence that is periodic or subperiodic, meaning that one of its exponents is in $\mathbb{Q}^+ \setminus \mathbb{N} = \{ 1, 2 \} \cup \{ 2, 3 \} \cup \{ 3, 4 \} \cdots$. The idea is to compute every possible factorization of $T = YZ$ into two factors, and try to (a) prolong each square sequence of $Q$ and try to (a) prolong each square sequence of $U$, or (b) extend $UU'$ to $UcU'$ for $U$ being a subsequence in $Y$, or (b) extend $UU'$ to $UcU'$ for $U'$ being a proper prefix of $U$ found in $Z$ and $Uc$ a subsequence found in $Y$. For a fixed partition $T = YZ$, we define

$D_2[y, z] := \max \begin{cases} 2 \cdot \text{LCS}_{Y, Z}(y - 1, z) + 1 & \text{if } \text{LCS}_{Y, Z}(y - 1, z - 1) > 0, \\ D_2[y - 1, z] + 1 & \text{if } D_2[y - 1, z] > 0, \\ D_2[y, z - 1], \\ 0, \end{cases}$

for $y \in [1..|Y|]$ and $z \in [1..|Z|]$ to be the longest subsequence of $T$ having an exponent in $\mathbb{Q}^+ \setminus \mathbb{N}$ of the form $UU'$ with $U'$ being a non-empty common subsequence of $Y[1..y]$ and $Z[1..y]$ and $U$ a subsequence of $Y[1..y]$, where $D_2[0, \cdot] := 0$. Note that we can add $D_2[y - 1, z - 1] + 1$ if $D_2[y - 1, z - 1] > 0$ as another selectable value for the maximum determining $D_2[y, z]$, but this does not change the maximum, since the value of $D_2[y - 1, z - 1] + 1$ is used as an option for the maximum determining $D_2[y, z - 1]$, which is already an option for $D_2[y, z]$. $D_2$ has the following property:

**Lemma 3.1.** $D_2[y, z]$ is the length of the longest subsequence of $T$ of the form $UU'$ with $U'$ being a proper prefix of $U$ and a common subsequence of $Y[1..y]$ and $Z[1..z]$, and $U$ being a subsequence of $Y[1..y]$.

**Proof.** Let $UU'$ be the longest subsequence having an exponent in $\mathbb{Q}^+ \setminus \mathbb{N}$ in $T$ with $Y[y_1] \cdots Y[y_{|U'|}] | Z[z_1] \cdots Z[z_{|V'|}] = UU'$. We partition $T = YZ$ such that $U$ and $U'$ are subsequences in $Y$ and $Z$, respectively. Since $U'$ is a proper prefix of $U$, $\text{LCS}_{Y, Z}(y_{|U'|}, z_{|V'|}) \geq |U'|$. If $\text{LCS}_{Y, Z}(y_{|U'|}, z_{|V'|}) > |U'|$, then there is another common subsequence $V'$ of $Y$ and $Z$, and $V' U[|U'|..]_+$ is a longer subsequence of $Y$ than $U$, so $V' U[|U'|..] U'$ is a longer subsequence having an exponent in $\mathbb{Q}^+ \setminus \mathbb{N}$ than $UU'$, a contradiction. Therefore, $\text{LCS}_{Y, Z}(y_{|U'|}, z_{|V'|}) = |U'|$. Obviously, $Y[y_{|U'|..}] = U[y_{|U'|..}]$, since otherwise we could extend $U$ further.

| Z  | a | b | c | a | a |
|----|---|---|---|---|---|
| Z[1] = a / aba abca abcaa abcaaa |
| Z[2] = c / aba abca acaac acaaac |
| Z[3] = a / aba abca acaac acaaac acaaaaca |
Finally, we show that the sequence $Y[y_1] \cdots Y[y_{|U'|}]Z[z_1] \cdots Z[z_{|U'|}]$ is considered in the construction of $D_2$. Since $y_{|U'|} + 1 \leq y_{|U'|+1} \leq |Y|$ (otherwise $U'$ cannot be a proper prefix of $U$), $D_2[y_{|U'|} + 1, z_{|U'|}] \geq 2 \cdot \text{LCS}_{Y,Z}(y_{|U'|}, z_{|U'|}) + 1$, and equality results from the fact that we otherwise have found a longer subsequence $VV'$ in $Y[y_{|U'|} + 1]Z[z_{|U'|}]$ that we can extend to a subsequence of $YZ$ longer than $UU'$ by applying the second option in Eq. (I). The third option of Eq. (I) fills up $D_2$ such that we obtain the length of $UU'$ in $D_2[|Y|, |Z|]$. □

Theorem 3.2. We can find the longest subsequence having an exponent in $\mathbb{Q}^+ \setminus \mathbb{N}$ in $O(n^3)$ time using $O(n^2)$ space.

Proof. For each of the $n$ different partitions $T = YZ$, we precompute a table answering LCS$_{Y,Z}(y, z)$ in constant time. This table needs $O(n^2)$ space and can be constructed in $O(n^2)$ time. Next, we create a table $D_2$ of size $O(n^2)$, and fill each of its cells by Eq. (I) in constant time thanks to the precomputation step. In total, we need $O(n^3)$ time per partition $T = YZ$, and therefore $O(n^3)$ time for the entire computation. □

In the following we want to omit subsequences having exponents only in $[1, 2]$.

4 Longest Periodic Subsequence

We now extend our ideas of the previous section to omit the sub-periodic subsequences, such that our algorithm always reports a subsequence with an exponent of at least 2. Our main idea is to generalize the factorization of $T$ from 2 to $k$ factors. Computing the LCS of these $k$ factors, we obtain all longest periodic subsequences having an exponent of length $k\ell$ with $\ell \in \mathbb{N}$. For $k = 2$, we can find all square subsequences, i.e., the longest common subsequence with an exponent of $2x$ for $x \in \mathbb{N}$, similar to [17]. Like in Sect. 3 we can support exponent values $\ell \in \mathbb{Q}^+$ by stopping matching characters in the last factor. In general, the number of factors $k = 3$ is a good value if the exponent is not in $(3, 4)$. With an exponent $x \in (3, 4)$, each factor starts capturing at the root of the repetition, i.e., if we match a subsequence $S = U^{|x|}U'$, then we start capturing $U$ in all factors simultaneously. However, the last factor has to capture $|UU'|$ characters if $x \in (3, 4)$. Hence, we need to split this last factor such that we have four factors, i.e., we need $k = 4$ for $x \in (3, 4)$.

However, $k = 3$ suffices for $x \in (2, 3] \cup [4, \infty)$. To see that, we let the first $k - 1$ factor capture the subsequence $U^{|x|/(k-1)}$. The last factor captures $U^y$ with $y = x - \lfloor x/(k-1) \rfloor (k-1)$, which works if $y \leq \lfloor x/(k-1) \rfloor$, i.e., $x \leq 3 \lfloor x/2 \rfloor$, which holds for $x \geq 4$. For $x \in (2, 3]$, each of the first factors captures $U$, while the last factor captures $U^y$ with $y \leq 1$.

4.1 Three Factors

There are $\binom{n}{3} = O(n^2)$ possible factorizations of the form $T = XYZ$. Let us fix one such factorization $T = XYZ$. For this factorization, we define the
Theorem 4.2. We can find the longest periodic subsequence with an exponent in \(3\mathbb{N}\) in \(O(n^3)\) time using \(O(n^3)\) space.

Proof. For the claimed time and space complexities, let us fix a factorization \(T = XYZ\). We first pre-process LCS with a three-dimensional table taking \(O(n^3)\) space and \(O(n^3)\) time such that we can answer an LCS query in \(O(1)\) time. The table \(D_3\) also takes \(O(n^3)\) space, and each cell can be filled in constant time thanks to the pre-processing step. Finally, we compute the maximum value of \(D_3\) for each factorization \(T = XYZ\), which are \(O(n^2)\) many. Hence, we fill \(D_3\) \(O(n^2)\) times, which gives us the total time of \(O(n^5)\).
4.2 Four Factors

Finally, we consider a factorization of size four to capture exponents in \((3, 4)\). We have \(\binom{4}{3} = \mathcal{O}(n^3)\) possibilities to factorize \(T\) into four factors \(W, X, Y, Z\). Let us fix a factorization \(T = W X Y Z\). We fill the 4-dimensional table \(D_4[1..|W|, 1..|X|, 1..|Y|, 1..|Z|]\) as follows:

\[
D_4[w, x, y, z] := \max \begin{cases} 
4 \cdot \text{LCS}_{W,X,Y,Z}(w - 1, x - 1, y - 1, z) + \delta_{wxy} & \text{if } \delta_{xyz} > 0 \\
D_4[w - 1, x - 1, y - 1, z] + \delta_{wxy} & \\
D_4[w - 1, x, y, z] & \\
D_4[w - 1, x, y, z] & \\
D_4[w, x, y - 1, z] & \\
D_4[w, x, y, z - 1] & \\
\end{cases}
\]

where \(\delta_{wxy} := 3\) if \(W[w] = X[x] = Y[y]\) and 0 otherwise.

**Theorem 4.3.** We can compute the longest periodic subsequence with an exponent \(\in \bigcup_{x \in \mathbb{N}} (3x, 4x) \cap \mathbb{Q}^+\) in \(\mathcal{O}(n^7)\) time using \(\mathcal{O}(n^4)\) space.

**Proof.** We can show analogously to Lemma 4.1 that \(D_4[w, x, y, z]\) stores the length of the longest string \(U^3U'\) for which \(U'\) is both a proper prefix of \(U\) and a subsequence of \(Z[1..z]\), while \(U\) is a common subsequence of the three strings \(W[1..w]\), \(X[1..x]\), and \(Y[1..y]\). The rest can be proven analogously to Thm. 4.2 by adding an additional dimension.

5 Open Problems

We are unaware of polynomial-time algorithms computing several other types of regularities when considering subsequences. For instance, we are not aware of an algorithm computing the longest \textit{sub-periodic} subsequence. For that, we would need an efficient algorithm computing the longest (common) subsequence \textit{without a border}. Then we could use the algorithm of Sect. 3 and compute this subsequence without a border instead of the (plain) LCS. Other problems are finding the longest (common) subsequence that is \textit{primitive} (exponent in \((1, \infty) \setminus \mathbb{N})\), or the longest (common) subsequence that is \textit{non-primitive} (exponent in \(\mathbb{N} \setminus \{1\}\)).

References

[1] R. A. Baeza-Yates. Searching subsequences. *Theor. Comput. Sci.*, 78(2): 363–376, 1991.

[2] H. Bannai, T. I, T. Kociumaka, D. Köppl, and S. J. Puglisi. Computing longest (common) Lyndon subsequences. *CoRR*, abs/2201.06773, 2022.
[3] T. C. Biedl, A. Biniaz, R. Cummings, A. Lubiw, F. Manea, D. Nowotka, and J. O. Shallit. Rollercoasters: Long sequences without short runs. *SIAM J. Discret. Math.*, 33(2):845–861, 2019.

[4] P. Bille, I. L. Gørtz, and F. R. Skjoldjensen. Subsequence automata with default transitions. *J. Discrete Algorithms*, 44:48–55, 2017.

[5] S. R. Chowdhury, M. M. Hasan, S. Iqbal, and M. S. Rahman. Computing a longest common palindromic subsequence. *Fundam. Informaticae*, 129(4):329–340, 2014.

[6] A. Conte, R. Grossi, G. Punzi, and T. Uno. Polynomial-delay enumeration of maximal common subsequences. In *Proc. SPIRE*, volume 11811 of *LNCS*, pages 189–202, 2019.

[7] L. Duraj. A sub-quadratic algorithm for the longest common increasing subsequence problem. In *Proc. STACS*, volume 154 of *LIPIcs*, pages 41:1–41:18, 2020.

[8] K. Fujita, Y. Nakashima, S. Inenaga, H. Bannai, and M. Takeda. Longest common rollercoasters. In *Proc. SPIRE*, volume 12944 of *LNCS*, pages 21–32, 2021.

[9] P. Gawrychowski, F. Manea, and R. Serafin. Fast and longest rollercoasters. In *Proc. STACS*, volume 126 of *LIPIcs*, pages 30:1–30:17, 2019.

[10] D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Commun. ACM*, 18(6):341–343, 1975.

[11] D. S. Hirschberg. Algorithms for the longest common subsequence problem. *J. ACM*, 24(4):664–675, 1977.

[12] S. Inenaga and H. Hyyrō. A hardness result and new algorithm for the longest common palindromic subsequence problem. *Inf. Process. Lett.*, 129:11–15, 2018.

[13] T. Inoue, S. Inenaga, H. Hyyrō, H. Bannai, and M. Takeda. Computing longest common square subsequences. In *Proc. CPM*, volume 105 of *LIPIcs*, pages 15:1–15:13, 2018.

[14] T. Inoue, S. Inenaga, and H. Bannai. Longest square subsequence problem revisited. In *Proc. SPIRE*, volume 12303 of *LNCS*, pages 147–154, 2020.

[15] M. Kiyomi, T. Horiyama, and Y. Otachi. Longest common subsequence in sublinear space. *Inf. Process. Lett.*, 168:106084, 2021.

[16] M. Kosche, T. Koß, F. Manea, and S. Siemer. Absent subsequences in words. In *Proc. RP*, volume 13035 of *LNCS*, pages 115–131, 2021.
[17] A. Kosowski. An efficient algorithm for the longest tandem scattered sub-
sequence problem. In *Proc. SPIRE*, volume 3246 of *LNCS*, pages 93–100,
2004.

[18] Y. Sakai. Maximal common subsequence algorithms. *Theor. Comput. Sci.*, 793:132–139, 2019.

[19] C. Schensted. Longest increasing and decreasing subsequences. *Canadian
Journal of Mathematics*, 13:179–191, 1961.

[20] S. Schrinner, M. Goel, M. Wulfert, P. Spohr, K. Schneeberger, and G. W.
Klau. The longest run subsequence problem. In *Proc. WABI*, volume 172 of *LIPIcs*, pages 6:1–6:13, 2020.