

Übungen zur Vorlesung Sequenzanalyse

Universität Bielefeld, WS 2025/26

Prof. Dr. Jens Stoye

Leonard Bohnenkämper

Tutorien: Lennart Finke, Sofie Jans

<https://gi.cebitc.uni-bielefeld.de/teaching/2025winter/sa1>

Übungsblatt 11 vom 22.01.2026

Abgabe bis 29.01.2026 bis 9:30 Uhr per Mail an den Tutor/die Tutorin

Aufgabe 1 (Burrows-Wheeler Transformation (BWT) – Naive Konstruktion) (5 Punkte)
Gegeben sei der String $s\$ = \text{AGTAGTAGTAG\$}$.

1. Berechne die Burrows-Wheeler Transformation $x = \text{BWT}(s\$)$ mithilfe der Rotationen von s . Mache dein Vorgehen deutlich.
2. Berechne das Run Length Encoding $\text{RLE}(x)$ von x . Fasse dabei nur Buchstaben zusammen, die mindestens dreimal hintereinander vorkommen.

Aufgabe 2 (Burrows-Wheeler Transformation (BWT) – LF-Mapping) (9 Punkte)

Gegeben sei die Sequenz $y = \text{EIIR(3G)NIE\$RINNRTA(3E)LHEER}$.

1. Dekomprimiere y in $z = \text{RLE}^{-1}(r)$.
2. y ist die Burrows-Wheeler Transformation einer Sequenz t . Berechne die erste Spalte F der zugehörigen Matrix.
3. Demonstriere die exakte Textsuche in der Burrows-Wheeler Transformation, indem du nach dem Pattern $p = \text{GEN}$ und dem Pattern $q = \text{FRIER}$ suchst. Das Suffix-Array von t ist dabei
25 4 24 7 9 1 21 11 14 8 0 20 18 3 23 13 17 5 10 2 22 6 19 12 15 16.
Wie oft kommen die Patterns jeweils vor? An welcher Position kommen sie vor? Gib alle Zwischen-schritte an.
4. Rekonstruiere die Sequenz $t\$ = \text{BWT}^{-1}(y)$. Mache auch hier dein Vorgehen deutlich.

Aufgabe 3 (Burrows-Wheeler Transformation (BWT) und Suffix-Array) (3 Punkte)
Implementiere eine Funktion, die die BWT aus dem Suffix-Array berechnet. Nutze dafür die vorgegebene Funktion `bwt_from_suffix_array` im Übungsrepository¹.

Aufgabe 4 (Burrows-Wheeler Transformation (BWT) und Approximative Textsuche) (3 Punkte)
Gegeben ein Pattern p und eine Sequenz s . Es soll ein Substring q von s ($s = rqt$) gefunden werden, sodass die Hamming-Distanz zwischen p und q eins oder weniger beträgt, $d_H(p, q) \leq 1$. Überlege dir einen Algorithmus, um einen solchen Substring mithilfe der Burrows-Wheeler Transformation $\text{BWT}(s)$ zu finden.

¹https://gitlab.ub.uni-bielefeld.de/lbohnenkaemper/sqa-ex/-/blob/main/w13/bwt.py?ref_type=heads