

Übungen zur Vorlesung Sequenzanalyse

Universität Bielefeld, SS 2023
Prof. Dr. Jens Stoye · Tizian Schulz

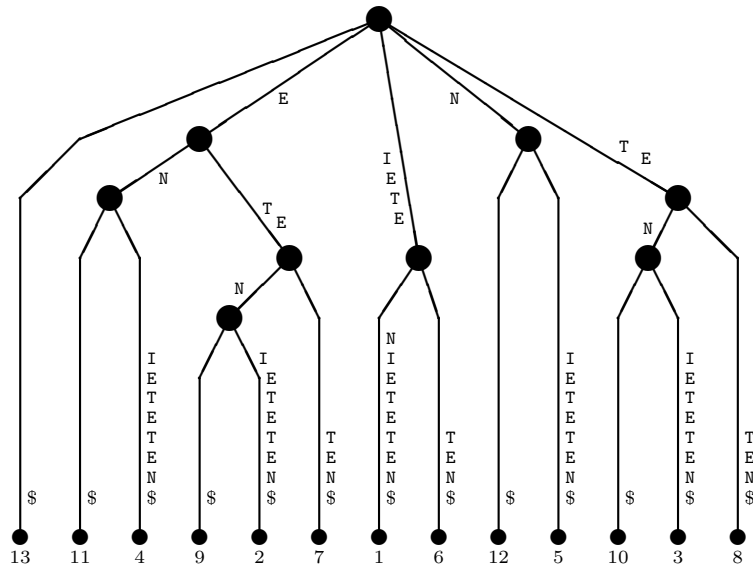
<https://gi.cebitec.uni-bielefeld.de/teaching/2023summer/sa>

Übungsblatt 9 vom 1.6.2023
Abgabe am 8.6.2023 bis 12:00 Uhr (mittags)

Aufgabe 1 (Suffixbaum und lcp-Array)

(3 Punkte)

Gegeben seien die Sequenz $s = \text{IETENIETETEN}$ und der Suffixbaum von $s\$$ mit lexikographisch sortierten Kanten:



Berechne mit Hilfe dieses Baums das lcp-Array von $s\$$, ohne `rank` oder `pos` explizit auszurechnen.

Aufgabe 2 (Manber-Myers-Algorithmus)

(9 Punkte)

1. Gegeben sei die Sequenz $s = \text{VTSVSVVTVTUTSVTSVSVSU}$.

- Erstelle das Suffix-Array `pos(s$)` unter Verwendung des Manber-Myers-Algorithmus. Gib das Zwischenergebnis nach jeder Phase des Algorithmus' an.
- Berechne die Arrays `rank(s$)` und `lcp(s$)`.
- Verwende Binärsuche im Suffix-Array `pos(s$)`, um alle Vorkommen des Musters $p = \text{SVS}$ in s zu finden. Gib alle Zwischenschritte an.

Aufgabe 3 (Manber-Myers-Algorithmus)

(3 Punkte)

Wie viele Phasen benötigt (im *worst case*) der Manber-Myers-Algorithmus für einen String der Länge n ? Gib eine Beispielsequenz an, bei der diese Anzahl Phasen erreicht wird.

Bitte wenden!

Aufgabe 4 (Burrows-Wheeler-Transformation)

(9 Punkte)

1. Gegeben sei der String $t = \text{ZZWYXZZWYXZZYXX}$.
 - (a) Berechne die Burrows-Wheeler-Transformierte $s = \text{BWT}(t\$)$. Gib alle Zwischenschritte an.
 - (b) Demonstriere den String-Matching-Algorithmus unter Verwendung der Sequenz $s = \text{BWT}(t\$)$ beispielhaft an der Suche des Musters $p = \text{WYXZZ}$ in t .
 - (c) Schreibe $r = \text{RLE}(s)$ als komprimierten String mit Hilfe von *run-length encoding* auf. Fasse dabei nur Buchstaben zusammen, die mindestens dreimal hintereinander vorkommen.
2. Gegeben sei der String $r = \text{XZ4YZW\$YW3ZXX}$.
 - (a) Dekomprimiere r in $s = \text{RLE}^{-1}(r)$.
 - (b) Rekonstruiere den String $t\$ = \text{BWT}^{-1}(s)$. Gib alle Zwischenschritte an.