

# Algorithmen zur Sequenzanalyse

Wintersemester 2017/2018  
Besprechung am 22.12.2017

## Übungsblatt 5

Prof. Dr. E. Ohlebusch,  
Institut für Theoretische Informatik

---

### Aufgabe 5.1.

Der String  $arr\$rcaaaabdb$  ist BWT eines Strings  $S$ . Rekonstruieren Sie  $S$ .

### Aufgabe 5.2.

Sei  $S$  ein String der Länge  $n$  ohne Abschlusszeichen  $\$$  und  $L$  die letzte Spalte der Matrix, die alle  $n$  Rotationen von  $S$  in lexikografischer Reihenfolge enthält. Sei  $SA$  das Suffix Array von  $S$  und der String  $BWT[1..n]$  definiert durch:

$$BWT[i] = \begin{cases} S[SA[i] - 1] & \text{falls } SA[i] > 1 \\ S[n] & \text{falls } SA[i] = 1 \end{cases}$$

Finden Sie einen String  $S$  für den  $L \neq BWT$  gilt.

### Aufgabe 5.3.

Erweitern Sie Algorithmus 15 des Skripts so, dass zusätzlich das Suffix Array des Strings berechnet wird. Ist es möglich, das LF-Array mit dem Suffix Array zu überschreiben um Speicherplatz zu sparen?

### Aufgabe 5.4.

Implementieren Sie die Move-To-Front-Transformation. Wenden Sie diese auf die Strings  $S = \text{diesisteintestdiesisteintest\$}$  und dessen BWT  $tt\$ttiittddeessiieeeeisssnn$  an. Vergleichen Sie die mittlere Huffman-Codewortlänge der beiden Move-To-Front-Transformierten. Testen Sie, ob sich in der Praxis ein Array oder eine Liste besser zur Verwaltung der Buchstabenliste eignet.

### Aufgabe 5.5.

Die Burrows-Wheeler Transformierte des Strings  $S = \text{in.ulm.um.ulm.und.um.ulm\$}$  ist  $BWT = \text{mmnmdmmn\$uuuluulliu.....}$ . Beantworten Sie mit Hilfe des Wavelet-Baums (siehe Abbildung 1) die folgenden Anfragen:

- $rank_i(BWT, 20)$
- $select_m(BWT, 4)$
- $BWT[17]$

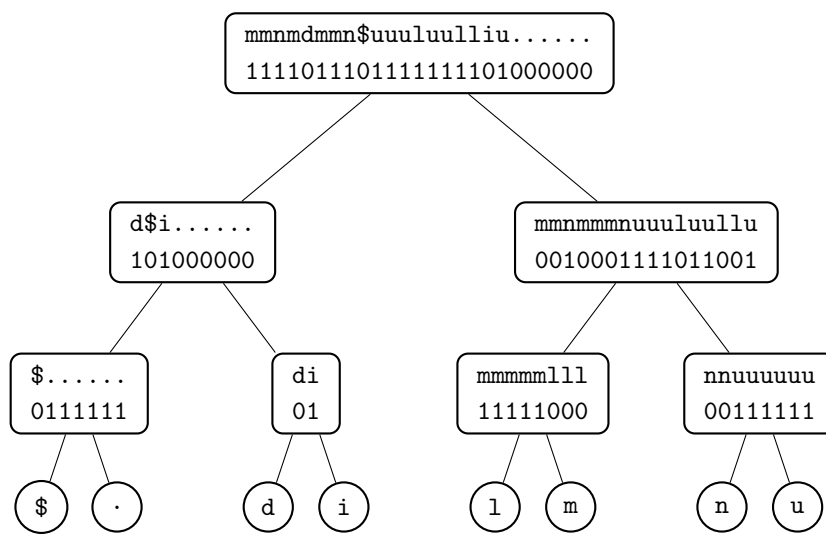


Abbildung 1: Wavelet-Baum des Strings *mmnmdmmn\$uuuluulliu.....*