

Algorithmen zur Sequenzanalyse

Wintersemester 2015/2016
Besprechung am 08.01.2016

Übungsblatt 5

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

Aufgabe 5.1.

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 gilt $L \neq BWT$.

Aufgabe 5.2.

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 5.3.

Geben Sie einen Algorithmus zur Konstruktion der Bitvektoren des Wavelet-Baums für den Fall $\sigma = 2^h$ in Pseudocode an. Analysieren Sie die Laufzeit und den Speicherbedarf Ihres Algorithmus.

Hinweis: Nummerieren Sie die Buchstaben des Alphabets von 0 bis $\sigma - 1$ und benutzen Sie die Binärdarstellungen der Nummern.

Aufgabe 5.4.

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

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

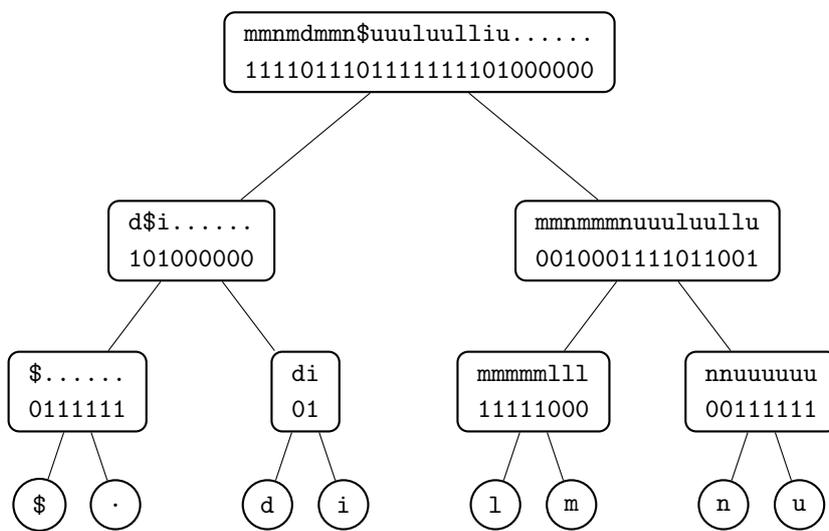


Abbildung 1: Wavelet-Baum des Strings *mmnmdmmn\$uuuluulliu.....*