Algorithmen zur Sequenzanalyse

Wintersemester 2014/2015 | **Übungsblatt 6** | Prof. Dr. E. Ohlebusch, T. Beller

Besprechung am 20.01.2015 | Institut für Theoretische Informatik

---

Aufgabe 6.1.
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.

Aufgabe 6.2.
Die Burrows-Wheeler Transformierte des Strings $S = in.ulm.um.ulm.und.um.ulm\$$ ist $BWT = mmnmdmmn\$uuuluulliu......$. Beantworten Sie mit Hilfe des Wavelet-Baums (siehe Abbildung 1) die folgenden Anfragen:

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

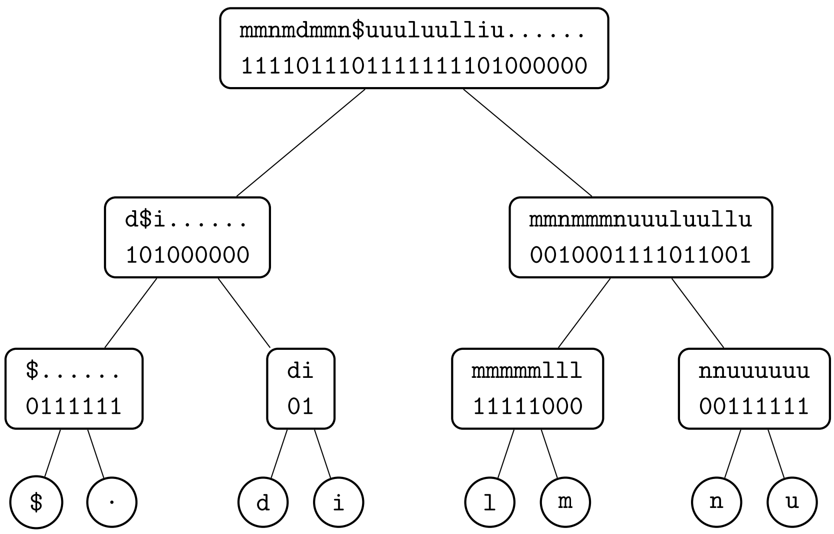

Abbildung 1: Wavelet-Baum des Strings *mmnmdmmn\$uuuluulliu......*

Aufgabe 6.3.
Abbildung 2 zeigt einen möglichen Huffman-Code-Baum für den String aus Aufgabe 6.2. Geben Sie einen (Huffman-Shaped) Wavelet-Baum an, welcher die gleiche Form wie der Huffman-Code-Baum hat. Beschreiben Sie, wie die Anfragen $BWT[i]$, $rank_c(BWT, i)$ und $select_c(BWT, i)$ auf diesem Wavelet-Baum beantwortet werden können. Welche Datenstrukturen werden dabei zusätzlich benötigt?

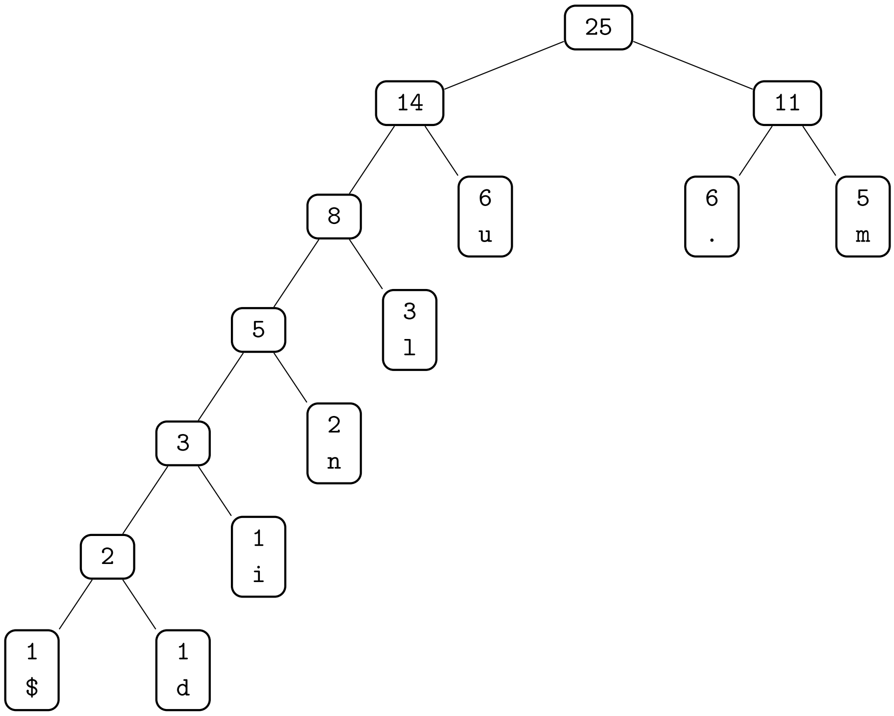

Abbildung 2: Möglicher Huffman-Code-Baum.

Aufgabe 6.4.

Ein Nachteil des Huffman-Shaped Wavelet-Baums (siehe Aufgabe 6.3) ist, dass dessen Tiefe nicht mit $\log \sigma$ beschränkt werden kann. Geben Sie die maximale Tiefe eines Huffman-Shaped Wavelet-Baums im Falle von $\sigma = 256$ an. Wie lang muss der zugehörige String mindestens sein, damit dieser Fall eintreten kann?

Aufgabe 6.5.

Zeigen Sie, dass $C[c]$ mit Hilfe des Wavelet-Baums der BWT in $\mathcal{O}(\log \sigma)$ berechnet werden kann. Warum wird man in der Praxis (z.B. bei Algorithmus 18 im Skript) dennoch das C-Array verwenden?