Algorithmische Grundlagen und Vermittlung der Informatik
Lehrstuhl 11 (Algorithm Engineering)
Prof. Dr. Johannes Fischer

technische universität dortmund

Text Indexing and Information Retrieval

Übungsblatt 7

Besprechung: 23.1.2014

Aufgabe 1 (Praxis)

Informieren Sie sich unter

`http://www-graphics.stanford.edu/~seander/bithacks.html#CountBitsSetNaive`

über die Möglichkeiten, in einem (32- oder 64-Bit) Wort die Anzahl der Einsen zu zählen. Implementieren Sie mit dieser Idee eine einfache rank-Datenstruktur, z.B. nur mit dem Array M und Block-Größe $s = 32$ oder $s = 64$ (je nachdem, wie Sie die Einsen in einem Wort zählen).

Aufgabe 2 (Theorie)

Diese Aufgabe baut auf Aufgabe 3 aus dem 6. Übungsblatt auf. Die Idee ist es, einen Wavelet-Tree mit der Topologie eines Huffman-Trees aufzubauen. Sei hierzu $L[1, n]$ ein Text über dem Alphabet Σ der Größe σ (z.B. eine BWT). Sei weiterhin $1 \leq n_c \leq n$ die Häufigkeit des Buchstabens c in T (für alle $c \in \Sigma$).

a) Beschreiben Sie, wie man einen Wavelet-Tree über L aufbauen kann, der die Suche nach einem Zeichen $L[i]$ in Zeit $O(\lg \frac{n}{n_c})$ unterstützt (also schnell für häufige Zeichen).

b) Analysieren Sie den Platz des resultierenden Wavelet-Trees. Gibt es Fälle, in denen die Datenstruktur kleiner als die $n \lg \sigma$ Bits eines „herkömmlichen“ Wavelet-Trees ist?