Übungen zur Vorlesung "Architektur und Programmierung von Grafik- und Koprozessoren"
Übungsblatt 8

Sommersemester 2019

8 Kompression

Aufgabe 8.1

Die nach ihren Erfindern benannte *Burrows Wheeler Transformation* (BWT) kommt in verschiedenen Anwendungen zum Einsatz, u. a. beim `bzip2` Kompressionsalgorithmus, z. B. aber auch beim Alignieren von Zeichenketten in der Bioinformatik. Zuerst hängt man der Zeichenkette ein spezielles Zeichen an (im Beispiel verwenden wir das Dollarzeichen ($)). Man transformiert nun die Zeichenkette, indem man zunächst eine Tabelle mit all ihren Rotationen bildet. Dann sortiert man alle Zeilen der Tabelle lexikographisch. Die letzte Spalte der sortierten Tabelle ist das Ergebnis der Transformation. Die BWT der Zeichenkette "EINSZWEI" erhält man z. B. wie folgt:

EINSZWEI$	EINSZWEI$
$EINSZWEI	EI$EINSZW
I$EINSZWE	INSZWEI$E
EI$EINSZW	I$EINSZWE
WEI$EINSZ	NSZWEI$EI
ZWEI$EINS	SZWEI$EIN
SZWEI$EIN	WEI$EINSZ
NSZWEI$EI	ZWEI$EINS
INSZWEI$E	$EINSZWEI

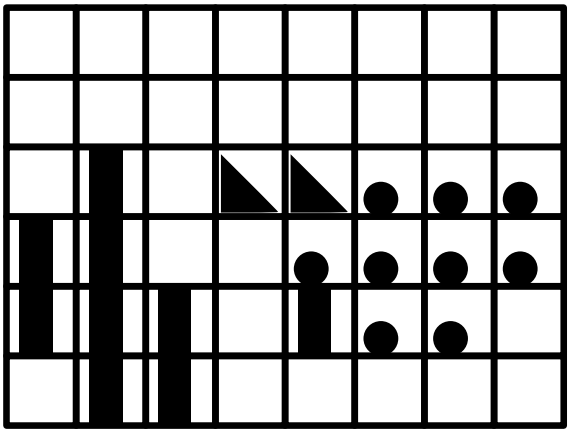
Sie lautet: "$WEEINZSI". Die BWT hat die Eigenschaften, dass sie Zeichenfolgen generiert, bei der häufig auftretende Zeichen tendentiell benachbart gespeichert werden, und dass die ursprüngliche Zeichenkette exakt aus der BWT rekonstruierbar ist. BWT Zeichenketten kann man daher z. B. gut mit Run-Length Encoding komprimieren.

Implementieren Sie die BWT mit Hilfe des Gerüstprogramms. Dieses verarbeitet ASCII Eingabedateien. Der Buchstabe $ ist reserviert und lexikographisch größer als alle anderen Buchstaben im Alphabet. Naiv implementiert benötigt der Algorithmus quadratisch viel Speicher in Relation zur Eingabe. Ihre Implementierung hingegen darf nur ein konstantes Vielfaches des Speicherbedarfs der Eingabezeichenkette benötigen. Anstatt jede Rotation einzeln zu speichern, speichern Sie nur die Indices, an denen die rotierten Worte beginnen, in einer Liste. Am Anfang sind die Indices aufsteigend sortiert. Anstatt nun beim Sortieren zwei Worte zu vertauschen, vertauscht man entsprechend die Position ihrer Startindices in der Liste der Indices. Die Reihenfolge in der Indexliste repräsentiert nun die korrekte lexikographische Sortierung der rotierten Zeichenketten.

Aufgabe 8.2

a.)
Gegeben das Alphabet $\Sigma = \{A, C, G, T\}$. Entwickeln Sie Huffman Codes für die Zeichenketten $AACCGGTTACGT$ sowie $AAAAAAAACGTA$.

b.)
Entwickeln Sie Huffman Codes für das Bild. Nehmen Sie an, dass das Bild über genau einen Farbkanal mit 2-bit Farbtiefe verfügt. Welche Kompressionsrate (Quotient aus unkomprimierter und komprimierter Größe) können Sie erzielen?



Das Übungsblatt wird am 27.06.2019 besprochen.