Exercises for the Lecture: “Architecture and Programming Models for GPUs and Coprocessors” Exercise Sheet № 7

Priv.-Doz. Dr. rer. nat. Stefan Zellmann

SS 2021

7 Compression

Ex. 7.1

The *Burrows Wheeler Transform* (BWT) is a building block of compression algorithms such as `bzip2`, but is also used, e.g., by gene sequence alignment algorithms in bioinformatics. The sequence / string to encode is first suffixed with a special token that is not part of the alphabet (in the example this is the dollar sign ($)). The string is transformed by first tabulating all the string's rotations. Then, the table is sorted *lexicographically.* The last column of the sorted table is the result of the transform. The tranform of the string “EINSZWEI” can for example be found as follows:

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

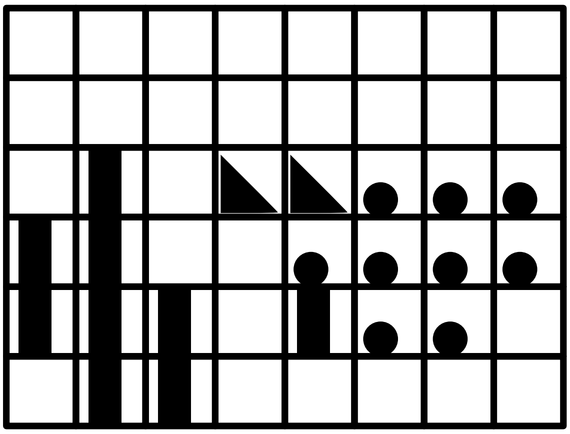
and has the value “$WEEINZSI”. BWT has the tendency to store same characters next to each other. Furthermore, the original string can be exactly reconstructed from the BWT. In practice, BWT is therefore often combined with run-length encoding.

Implement the BWT using the skeleton program. The skeleton program processes ASCII input files. The character $ is reserved and lexicographically greater than the other characters in the alphabet. While a naïve implementation will use $O(n^2)$ memory in the length of the input sequence n, your implementation's memory requirement is supposed to be $O(n)$. Instead of storing each rotation in its own table entry, you use a list of indices pointing into the original string. The indices are associated with the rotations' first characters. At the beginning, the indices are sorted in ascending order. While sorting, instead of swapping table entries, you instead swap start indices in the index list. The order of the indices will then eventually represent the correct lexicographic order of the rotated string.

Ex. 7.2

a.)
Given the alphabet $\Sigma = \{A, C, G, T\}$, construct the Huffman codes for the text strings $AACCGGTTACGT$ as well as $AAAAAAAACGTA$.

b.)
Construct Huffman codes for the image. You can assume that the image is encoded with a single 2-bit color channel. What is the compression rate that you can achieve?



The exercise sheet will be discussed on July 01, 2021.