Technische Universität München
Fakultät für Informatik
Lehrstuhl für Theoretische Informatik
Prof. Dr. Susanne Albers
Jens Quedenfeld

Winter Semester 2017/18
Problem Set 6
November 27, 2017

Online and Approximation Algorithms

Due December 11, 2017 at 10:00

Exercise 1 (Compression – 13 points)

Consider the alphabet $\Sigma = \{\mathrm{i}, \mathrm{m}, \mathrm{p}, \mathrm{s}\}$ and the string $S =$ 'missmississippi' over Σ.

(a) Calculate the Burrows-Wheeler transformation of S.

(b) Show the encoding procedure of the transformed text of (a) by using Linear List compression and the Move-To-Front algorithm, assuming that the initial list is $(\mathrm{i}, \mathrm{m}, \mathrm{p}, \mathrm{s})$. Show all intermediate steps.

(c) Encode the result of (b) using the Huffman code. Include the resulting tree, the encoding table as well as the encoded bit string.

Exercise 2 (Decompression – 10 points)

Consider the alphabet $\Sigma = \{\mathrm{a}, \mathrm{b}, \mathrm{n}, \mathrm{s}\}$. An unknown string S was transformed with the Burrows-Wheeler transformation, afterwards encoded with Linear List compression using the Move-To-Front algorithm (the initial list was sorted alphabetically) and then compressed with the Huffman code. The result of this procedure is

0110 1010 1100 0111 11

The Huffman table is

$$00 \to 4$$
$$01 \to 3$$
$$1 \to 1$$

The index of the Burrows-Wheeler transformation that contains the original string is $I = 6$.
Find the original string S. Execute the Burrows-Wheeler back transformation with linear space.

Exercise 3 (Huffman code – 7 points)

Let $P = (p_1, \ldots, p_n)$ be an memoryless source over an alphabet $\Sigma = \{x_1, \ldots, x_n\}$. In the literature there is sometimes the claim that the number of bits of the Huffman code of the symbol x_i is *never* greater than $\lceil -\log_2 p_i \rceil$. However, this is not the case. Find an example.

Exercise 4 (Prefix code properties – 10 points)

Let C^* be an arbitrary optimal prefix code where the symbol x_i that occurs with probability of $p_i > 0$ is encoded by l_i bits.

(a) Prove: if $p_j > p_k$, then $l_j \leq l_k$

(b) Prove: The two symbols with the lowest probability are encoded by the two longest code words which have the same length.

In the exercise session we will prove the optimality of Huffman codes as an additional presence exercise. You do not need to do this at home!