UNIVERSITY OF LIVERPOOL

Department of Computer Science
Dr. Sebastian Wild · Ben Smith

Date: 2022-11-23
Version: 2022-11-23 09:13

# Tutorial 8 for COMP 526 – Efficient Algorithmics, Fall 2022

## Problem 1 (Huffman code)

Compress the text $T =$ `HANNAHBANSBANANASMAN` using a Huffman code; give

1. the character frequencies,
2. a step-by-step construction of the Huffman tree,
3. the Huffman code, and
4. the encoded text.
5. Finally, compute the compression ratio of the result (ignoring space needed to store the Huffman code).

## Problem 2 (No Free Lunch)

Prove the following *no-free-lunch* theorems for lossless compression.

1. *Weak version:* For every compression algorithm $A$ and $n \in \mathbb{N}$ there is an input $w \in \Sigma^n$ for which $|A(w)| \ge |w|$, i. e. the "compression" result is no shorter than the input.

   **Hint:** Try a proof by contradiction. There are different ways to prove this.

2. *Strong version:* For every compression algorithm $A$ and $n \in \mathbb{N}$ it holds that

$$\bigl|\{w \in \Sigma^{\le n} : |A(w)| < |w|\}\bigr| \;<\; \tfrac{1}{2} \cdot \bigl|\Sigma^{\le n}\bigr| .$$

   In words, less than half of all inputs of length at most $n$ can be compressed below their original size.

   **Hint:** Start by determining $|\Sigma^{\le n}|$.

The theorems hold for every non-unary alphabet, but you can restrict yourself to the binary case, i.e., $\Sigma = \{0, 1\}$.

We denote by $\Sigma^\star$ the set of all (finite) strings over alphabet $\Sigma$ and by $\Sigma^{\le n}$ the set of all strings with size $\le n$. As domain of (all) compression algorithms, we consider the set of (all) *injective* functions in $\Sigma^\star \to \Sigma^\star$, i.e., functions that map any input string to some output string (encoding), where no two strings map to the same output.