Stefan Walzer, Stefan Hermann
ITI Prof. Sanders
winter term 2025/2026

Exercise Sheet 16 – Retrieval

Probability and Computing

Exercise 1 – AMQ from Retrieval

Let $b \in \mathbb{N}$ and S be a set of size $n = |S|$. Use the peeling-based retrieval data structure from the lecture as a black box to construct a static filter (i.e., an approximate membership query data structure) for S with false-positive probability $\varepsilon = 2^{-b}$.

State advantages and disadvantages of the resulting data structure compared to a Bloom filter with the same false-positive probability. You may assume that $b = \mathcal{O}(\log n)$, so that bit strings of length b can be processed in time $\mathcal{O}(1)$.

Exercise 2 – Learned Data Structures

Let S be a set of $n = |S|$ names with a uniquely associated gender $f : S \longrightarrow \{\text{F}, \text{M}\}$. A clever student observes that most $x \in S$ with $f(x) = \text{F}$ end in a vowel and most $x \in S$ with $f(x) = \text{M}$ end in a consonant. This simple rule works for all but δn of the names, for a small $\delta > 0$.

Construct a data structure with expected space usage $\mathcal{O}(\delta n \log(1/\delta))$ that returns the correct gender $f(x)$ for every $x \in S$.

Hint: Combine an AMQ filter and a retrieval data structure in a clever way.

Remark: By *learned data structures* one understands a combination of classical data structures and machine learning techniques. As indicated by this exercise, the idea is to beneficially combine the pattern-recognition capabilities of machine learning techniques with the reliability guarantees of classical data structures.

Exercise 3 – Retrieval with Variable Bit Length

According to the lecture, for every universe D, every set $S \subseteq D$, and every function $f : S \longrightarrow \{0, 1\}$ we can construct a retrieval data structure for f with space usage $1.23|S|$. This shall be used here as a black box.

Construct a retrieval data structure for the case in which the range of the function f contains bit strings of variable length.

More precisely, let $C \subseteq \{0, 1\}^*$ be a prefix-free code, D' a universe, $T \subseteq D'$, and $g : T \longrightarrow C$ a function. Construct a data structure R with space usage $1.23 \cdot \sum_{x \in T} |g(x)|$ and a corresponding algorithm eval such that for every $x \in T$ we have $\text{eval}(R, x) = g(x)$.

Hint: Introduce as many keys for each $x \in T$ as the length $|g(x)|$ of $g(x)$.