

# Übungsblatt 16 – Retrieval

## Randomisierte Algorithmik

### Aufgabe 1 – AMQ aus Retrieval

Sei  $b \in \mathbb{N}$  und  $S$  eine Menge der Größe  $n = |S|$ . Verwende die Peeling-basierte Retrieval Datenstruktur der Vorlesung als Black-Box um einen statischen Filter (also eine Approximate-Membership-Query Datenstruktur) für  $S$  mit falsch-positiv Wahrscheinlichkeit  $\varepsilon = 2^{-b}$  zu konstruieren.

Nenne Vor- und Nachteile der entstandenen Datenstruktur im Vergleich zu einem Bloom-Filter mit gleicher falsch-positiv Wahrscheinlichkeit. Du darfst annehmen, dass  $b = O(\log n)$ , sodass Bistrings der Länge  $b$  in Zeit  $O(1)$  verarbeitet werden können.

### Aufgabe 2 – Learned Data Structures

Sei  $S$  eine Menge von  $n = |S|$  Namen mit eindeutig zuordenbarem Geschlecht  $f : S \rightarrow \{F, M\}$ . Eine findige Studentin bemerkt, dass die meisten  $x \in S$  mit  $f(x) = F$  auf einen Vokal enden und die meisten  $x \in S$  mit  $f(x) = M$  auf einen Konsonanten enden. Diese simple Regel funktioniert für alle außer  $\delta n$  der Namen, für ein kleines  $\delta > 0$ .

Konstruiere eine Datenstruktur mit erwartetem Speicherbedarf  $O(\delta n \log(1/\delta))$ , die für jedes  $x \in S$  das korrekte Geschlecht  $f(x)$  liefert.

**Hinweis:** Stöpsle dazu einen AMQ-Filter und eine Retrieval Datenstruktur geschickt zusammen.

**Bemerkung:** Unter *Learned Data Structures* versteht man eine Kombination aus klassischen Datenstrukturen und Machine Learning Techniken. Wie in dieser Aufgabe angedeutet, geht es darum, die Mustererkennungsfähigkeiten von Machine Learning Techniken mit den Verlässlichkeitsgarantien klassischer Datenstrukturen nutzbringend zu verheiraten.

### Aufgabe 3 – Retrieval mit Variabler Bitlänge

Gemäß Vorlesung können wir für jedes Universum  $D$ , jede Menge  $S \subseteq D$  und jede Funktion  $f : S \rightarrow \{0, 1\}$  eine Retrieval-Datenstruktur für  $f$  mit Speicherbedarf  $1.23|S|$  konstruieren. Dies soll hier als Black-Box verwendet werden.

Konstruiere eine Retrieval Datenstruktur für den Fall in dem der Wertebereich der Funktion  $f$  Bitstrings variabler Länge enthält.

Genauergesagt, sei  $C \subseteq \{0, 1\}^*$  ein präfixfreier Code,  $D'$  ein Universum,  $T \subseteq D'$  und  $g : T \rightarrow C$  eine Funktion. Konstruiere eine Datenstruktur  $R$  mit Speicherbedarf  $1.23 \cdot \sum_{x \in T} |g(x)|$  und einen zugehörigen Algorithmus eval sodass für jedes  $x \in T$  gilt  $\text{eval}(R, x) = g(x)$ .

**Hinweis:** Führe für jedes  $x \in T$  so viele Schlüssel ein, wie die Länge  $|g(x)|$  von  $g(x)$  beträgt.