

Text Indexing and Information Retrieval

Übungsblatt 1

Besprechung: 23./24.10. 2013

Aufgabe 1 (Praxis)

Implementieren Sie einen Trie, sodass damit Strings lexikographisch sortiert werden können. Verwenden Sie dabei eine sinnvolle Repräsentation der Trie-Knoten aus der Vorlesung. Es bietet sich beispielsweise an, die ausgehenden Kanten zunächst unsortiert zu halten (z.B. mit C++ `hash_map`), und dann, nach dem Einfügen aller Strings, diese zu sortieren.

Testen Sie die Laufzeit mit dem auf der Webseite zur Verfügung gestellten Strings und vergleichen Sie sie mit `gnu sort`.

Aufgabe 2 (Theorie)

Möglichkeit (5) zur Repräsentation von Trie-Knoten war, gewichtsbalancierte binäre Suchbäume zu verwenden, bei denen die Zeiger zu den Kindern im Trie an den Blättern gespeichert waren. Suchbäume können ihre Information (hier also die Zeiger auf die Trie-Knoten) aber auch an internen Knoten speichern. Beschreiben Sie ein entsprechendes Verfahren für die gewichtsbalancierten binären Suchbäume an den Trie-Knoten (wie wird der Suchbaum für einen Knoten v mit Kindergewichten w_{v_1}, \dots, w_{v_x} aufgebaut?) und zeigen Sie, dass dies ebenfalls zu einer Suchzeit von $O(|P| + \log k)$ führt.

Dies ist in der Tat einfacher als bei den in der Vorlesung verwendeten Suchbäumen, bei denen die Information nur an den Blättern stand!