Algorithmische Grundlagen und Vermittlung der Informatik
Lehrstuhl 11 (Algorithm Engineering)
Prof. Dr. Johannes Fischer

technische universität dortmund

Text Indexing and Information Retrieval

Übungsblatt 3

Besprechung: 7.11.2016

Aufgabe 1 (Theorie+Praxis)

a) Führen Sie den in der Vorlesung besprochenen Linearzeit-Algorithmus für Suffix Arrays anhand des Beispieltextes $T =$ `antananarivo$` aus.

b) Die folgende Grammatik erzeugt einen Text der Länge $n = 2^p$ über dem Alphabet $\{\$, \sigma_1, \ldots, \sigma_p\}$: $S \to T_1\$$, $T_i \to T_{i+1}\sigma_i T_{i+1}$ für $i = 1, \ldots, p-1$ und $T_p \to \sigma_i$. Führen Sie den in der Vorlesung besprochenen Linearzeit-Algorithmus für Suffix Arrays auf dem aus $p = 4$ resultierenden Text S aus.

c) Was ist das Besondere an dem in Aufgabenteil (b) erzeugten Text in Hinsicht auf den in der Vorlesung besprochenen Linearzeit-Algorithmus für Suffix Arrays?

Aufgabe 2 (Praxis)

Auf der Seite `https://sites.google.com/site/yuta256/sais` befinden sich gute Implementierungen des in der Vorlesung besprochenen Linearzeit-Algorithmus für Suffix Arrays. Stellen Sie diesen Code mit seinen Implementierungstricks vor (Hauptdateien `sais.c` bzw. `sais.java`).

Aufgabe 3 (Praxis)

Lassen Sie den Algorithmus aus Aufgabe 2 auf möglichst großen Texten von der Seite `http://pizzachili.dcc.uchile.cl/texts.html` laufen und messen Sie Laufzeiten und Speicherplatzbedarf. Wenn möglich, vergleichen Sie die Laufzeit und den Speicherbedarf mit Ihrem in Übungsblatt 2 programmierten $O(n \lg n)$-Zeit Algorithmus für Suffix-Arrays.

Aufgabe 4 (Theorie)

Beschreiben Sie einen Algorithmus in Pseudocode, der die S*-substrings eines Textes $T[1, n]$ mittels Radix-Sort in Linearzeit sortiert (nehmen Sie hierfür an, dass die Alphabetgröße σ höchstens n ist).