Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Dr. Riko Jacob
Matthias Baumgart

Sommersemester 2009
Übungsblatt 7
30. Juni 2009

Effiziente Algorithmen und Datenstrukturen II

Abgabetermin: 07.07.2009 **vor** *der Vorlesung*

Aufgabe 1 (1 Punkt)

Geben Sie für alle $n \in \mathbb{N}$ einen Text und ein Suchwort an, so dass die naive Textsuche mit Hilfe eines Suffix-Arrays (ohne LCP-Array) Laufzeit $\Omega(n \log n)$ hat.

Aufgabe 2 (2 Punkte)

Wie kann man aus einem Suffix-Baum möglichst schnell ein Suffix-Array sowie das entsprechende LCP-Array konstruieren? Beschreiben Sie dafür einen effizienten Algorithmus. Welche Voraussetzungen muss der Suffix-Baum dabei erfüllen?

Aufgabe 3 (3 Punkte)

Ein Teilwort $s = s_1 \dots s_l$ eines Wortes $t \in \Sigma^n$ heißt minimal-rechtseindeutig, wenn es an genau einer Position in t als Teilwort vorkommt und wenn das Präfix $s_1 \dots s_{l-1}$ von s mindestens zweimal als Teilwort von t auftritt.

Entwerfen Sie einen Algorithmus, der in Linearzeit für ein Wort $t \in \Sigma^*$ alle seine minimal-rechtseindeutigen Teilwörter findet.

Aufgabe 4 (4 Punkte)

Gegeben sei ein endliches Alphabet Σ sowie ein sogenanntes *dont-care*-Symbol $* \notin \Sigma$. Ein Wort $s \in (\Sigma \cup \{*\})^m$ ist Teilwort von $t \in \Sigma^n$, falls es ein $i \in [0 \dots n-m]$ gibt, so dass für alle $j \in [0 \dots m-1]$ entweder $s_j = *$ oder $s_j = t_{i+j}$ gilt.

Entwerfen Sie einen effizienten Algorithmus, der auch Suchworte findet, die dont-care-Symbole enthalten. Welche Laufzeit hat Ihr Algorithmus in Abhängigkeit von n, m und der Anzahl k der dont-care-Symbole in s?