Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Volker Heun

Wintersemester 2018/19
Übungsblatt 11
10. Januar 2019

Algorithmen auf Sequenzen

Abgabetermin: Donnerstag, den 17. Januar vor der Vorlesung

Aufgabe (Notenbonus) 1

Sei $t = w_1\# \cdots \# w_m \in (\Sigma \cup \{\#\})^*$ mit $w_i \in \Sigma^+$ und sei $n = |t|$.

Beschreibe einen Algorithmus, der t in Zeit $O(n)$ so vorverarbeitet, dass anschließend für jedes Wort p in Zeit $O(|p|)$ entschieden werden kann, ob p eines der w_i ist, wobei der zusätzlich zum Text t benötigte Platz der resultierenden Datenstruktur $O(m)$ sein soll (wobei die Zeichenreihe t selbst auch zur Verfügung steht).

Wie groß ist der (maximale) zusätzliche Platzbedarf der vorgeschlagenen Datenstruktur während der Konstruktion?

Aufgabe (Notenbonus) 2

Zeige, wie man für den in der Vorlesung vorgestellten Algorithmus zum Suchen in Suffix-Arrays für $t \in \Sigma^n$ mittels binärer Suche in Zeit $O(m + \log(n))$ die benötigten lcp-Werte in einer Vorverarbeitung mit einem Zeitbedarf $O(n)$ und Platzbedarf $O(n)$ unabhängig vom Suchwort $s \in \Sigma^m$ vorab berechnen kann, so dass jede lcp-Anfrage bei der eigentlichen Suche nach s in konstanter Zeit beantwortet werden kann. Hierbei soll keine RMQ-Datenstruktur verwedet werden.

HINWEIS: Das Feld L mit $L[i] = \text{lcp}(i-1, i)$ darf verwendet werden.

Aufgabe 3

Sei $t \in \Sigma^*$ ein Text und $k \in \mathbb{N}$. Wie kann in $O(|t|)$ Zeit festgestellt werden, wie viele verschiedene Teilstrings der Länge k in t enthalten sind? Gib hierzu einen Algorithmus in Pseudo-Code an.

Wir wünschen allen
ein erfolgreiches und gesundes
neues Jahr 2019!