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

Wintersemester 2017/18
Übungsblatt 11
18. Januar 2018

Algorithmen auf Sequenzen

Abgabetermin: Donnerstag, den 25. Januar vor der Vorlesung

Aufgabe (Notenbonus) 1

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.

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

Aufgabe (Notenbonus) 2

In der Vorlesung wurde gezeigt, wie man mit Hilfe eines Suffix-Arrays für $t \in \Sigma^n$ in Zeit $O(m+\log(n))$ feststellen kann, ob t ein Wort $s \in \Sigma^m$ enthält oder nicht. Modifiziere diesen Algorithmus so, dass er alle z Vorkommen von s in t mit Zeitbedarf $O(m + \log(n) + z)$ findet.

Wie kann man den Zeitbedarf auf $O(m + \log(n))$ senken?

Gib jeweils einen Algorithmus im Pseudo-Code an.

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.