
Algorithmen auf Sequenzen

Abgabetermin: Donnerstag, den 24. Januar vor der Vorlesung

Aufgabe (Notenbonus) 1

Konstruiere für das Wort $t = aaabaabab$ das zugehörige Extended-Suffix-Array, d.h. das Suffix-Array A und die LCP-Tabelle L und zeichne den zugehörigen LCP-Intervall-Baum von t (es gilt $\$ < a < b$).

Aufgabe (Notenbonus) 2

Betrachte das Wort $t = \text{ABANANAANDANANANAS}$.

- Konstruiere die Burrows-Wheeler-Transformierte \hat{t} zu $t\$$.
- Gib die zugehörige LF-Funktion für \hat{t} an.
- Gib die Werte $C(\cdot)$ und $Occ(\cdot, \cdot)$ an.
- Suche nach $s = \text{NANA}$ im FM-Index für t mit Hilfe des in der Vorlesung angegebenen Algorithmus für die Rückwärtssuche im FM-Index.

Es gilt $\$ < A < B < D < N < S$.

Aufgabe 3

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.

*Für die Teilnahme an der
Klausur am 13. Februar
ist auch eine Anmeldung
über TUMonline erforderlich.*