

Algorithmen zur Sequenzanalyse

Wintersemester 2019/2020
Besprechung am 29.10.2019

Übungsblatt 1

Prof. Dr. E. Ohlebusch,
Institut für Theoretische Informatik

Aufgabe 1.1.

Berechnen Sie mit Hilfe des Skew-Algorithmus das Suffixarray des Strings *mississippi*.

Aufgabe 1.2.

Im Skript wurden die Funktionen τ und τ^{-1} (für $n \bmod 3 = 0$) wie folgt definiert:

$$\tau(i) = \begin{cases} \frac{i+1}{3}, & \text{falls } i \bmod 3 = 2 \\ \frac{n+i}{3}, & \text{falls } i \bmod 3 = 0 \end{cases}$$
$$\tau^{-1}(j) = \begin{cases} 3j - 1, & \text{falls } 1 \leq j \leq \frac{n}{3} \\ 3j - n, & \text{falls } \frac{n}{3} < j \leq \frac{2n}{3} \end{cases}$$

Zeigen Sie, dass τ^{-1} die Umkehrfunktion von τ ist, also dass gilt: $\tau(\tau^{-1}(i)) = i$

Aufgabe 1.3.

Beweisen Sie, dass für $i \neq j$ mit $i \bmod 3 \neq 1$ und $j \bmod 3 \neq 1$ gilt:

$$S_i < S_j \quad \text{gdw.} \quad \overline{S}_{\tau(i)} < \overline{S}_{\tau(j)}$$

Hinweis: Erweitern Sie die lexikografischen Namen $\overline{S}_{\tau(i)}$ und $\overline{S}_{\tau(j)}$ um die zugehörigen Tripel und benutzen Sie eine Fallunterscheidung für $i \bmod 3$ und $j \bmod 3$.

Aufgabe 1.4.

Gegeben sei ein String S der Länge n sowie sein Suffixarray. Geben Sie einen Algorithmus an, der die Anzahl der Vorkommen eines Musters P im String S in $O(m \log n)$ Zeit berechnet, wobei m die Länge von P ist.

Beispiel: Das Muster $P = taat$ kommt zweimal im String $S = ctaataatg$ vor.