

Algorithmische Bioinformatik I

Aufgabe 1

Entwerfen Sie einen effizienten Algorithmus, um das längste Palindrom, das zugleich auch das Suffix eines gegebenen Textes t der Länge n ist, zu finden. Ein Palindrom p der Länge m hat die Form $p_0 \dots p_{m-1} = p_{m-1} \dots p_0$. Analysieren Sie die Laufzeit des Algorithmus.

Aufgabe 2

Ein zyklischer String S ist ein unendlicher String, für den gilt $\exists i : \forall j : s_j = s_{j+i}$. Das kleinste solche i ist die Länge des zyklischen Strings. Gegeben seien zwei zyklische Strings S_1 mit Länge l_1 und S_2 mit Länge l_2 , $l_1 \leq l_2$. Geben Sie einen möglichst effizienten Algorithmus an, der herausfindet, ob ein Substring von S_1 der Länge l_1 in S_2 enthalten ist. Was ist die Laufzeit des Algorithmus?

Aufgabe 3

Sei Σ ein endliches Alphabet und sei $*$ $\notin \Sigma$ ein so genanntes *dont-care*-Symbol. Ferner sei $s \in (\Sigma \cup \{*\})^+$ und $t \in \Sigma^+$ mit $|s| = m$ und $|t| = n$ sowie $k \in \mathbb{N}$ eine Konstante, die unabhängig von n und m ist.

Entwerfen Sie einen Algorithmus, der feststellt, ob s ein Teilwort von t *matched*, d.h., ob es ein $i \in [0 : n - m]$ gibt, so dass für alle $j \in [0 : m - 1]$ entweder $s_j = *$ oder $s_j = t_{i+j}$ ist. Wenn das *dont-care*-Symbol dabei genau k -mal in s auftritt, soll der Algorithmus dabei maximal $O(k \cdot n + m)$ Vergleiche ausführen.

Hinweis: Versuchen Sie den Algorithmus von Knuth-Morris-Pratt geeignet zu modifizieren.

Aufgabe 4

Auf der rechten Seite ist der Suffix-Trie für das Wort $t = ababba$ angegeben. Konstruieren Sie aus diesem mit Hilfe des in der Vorlesung angegebenen On-Line-Algorithmus schrittweise die Suffix-Tries für

- $t' := t \cdot b = ababbab$,
- $t'' := t' \cdot a = ababbaba$ und
- $t''' := t'' \cdot a = ababbabaa$.

Man gebe hierzu für jedes Wort t' , t'' und t''' eine eigene Zeichnung mitsamt der verwendeten und neu konstruierten Suffix-Links an.

