
Algorithmische Bioinformatik I

Aufgabe 1

- Betrachten Sie einen Suffix-Baum. Sei $f()$ die Suffix-Link-Funktion und \bar{v} ein Knoten im Suffixbaum. Zeigen Sie, dass \bar{v} ist kein Blatt auch $f(\bar{v})$ ist kein Blatt impliziert.
- Beweisen Sie, dass die Größe eines Suffixbaums für einen String der Länge n $\Theta(n^2)$ im Worst Case ist, falls die Kantenlabel explizit ausgeschrieben werden.

Aufgabe 2

- Man entwerfe einen Algorithmus für das Problem *exact string matching* auf der Basis von Suffixbäumen, der zu einem Text t und einem Suchwort s die Position *jedes* Vorkommens von s in t ausgibt. Für k Vorkommen von s in t soll die Komplexität des Algorithmus $O(|s| + |t| + k)$ betragen.
- Wie kann man die Komplexität der Suche nach s auf $O(|s|)$ senken, wenn nur höchstens ein Vorkommen von s in t auszugeben ist?
- Wie groß ist der asymptotische Platzbedarf einer optimal implementierten Datenstruktur für einen Suffixbaum eines Textes der Länge n ?

Aufgabe 3

- Entwirf eine Methode zur Konstruktion eines so genannten *verallgemeinerten Suffixbaumes*, der die Suffixe aller Worte der Menge $M = \{w_1, w_2, \dots, w_n\}$ mit $w_i \in \Sigma^*$ von Zeichenreihen beinhaltet.

Hinweis: Betrachte dazu das Wort $w' = w_1\$1w_2\$2 \cdots w_n\$n$, wobei $\$, \dots, \$n \notin \Sigma$ neue Symbole sind.

- Dieser Baum kann jedoch größer als notwendig sein. Verfeinere ggf. die Methode zur Erzeugung eines verallgemeinerten Suffixbaumes, so dass der verallgemeinerte Suffixbaum nur noch alle Suffixe der Wörter von M enthält.
- Konstruiere nach Deiner Methode den Suffixbaum für die Menge

$\{ababbab, aabab, babab, abba\}$.

Aufgabe 4

Gegeben sei eine Menge $S = \{s_1, \dots, s_\ell\}$ von Zeichenreihen mit $n = \sum_{i=1}^{\ell} |s_i|$. Konstruiere einen Algorithmus, mit dem man in Zeit $O(n)$ alle Zeichenreihen $s_i \in S$ finden kann, die Teilwörter einer anderen Zeichenreihe $s_j \in S$ sind.