Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Dr. Hanjo Täubig
Johannes Nowak

Sommersemester 2008
Übungsblatt 4
09. Juni 2008

Algorithmische Bioinformatik I

Aufgabe 1

a) 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.

b) Wie kann man die Komplexität der Suche nach s auf $O(|s|)$ senken, wenn nur höchstens ein Vorkommen von s in t auszugeben ist?

c) Welchen Platz belegt eine optimal implementierte Datenstruktur für einen Suffixbaum eines Textes der Länge n?

Aufgabe 2

a) Konstruiere eine Methode zur Konstruktion eines so genannten *verallgemeinerten Suffixbaumes* an, der die Suffixe aller Worte der Menge $M = \{w_1, w_2, \ldots, w_n\}$ von Zeichenreihen beinhaltet, wobei $w_i \in \Sigma^*$.

   Hinweis: Betrachte dazu das Wort $w' = w_{11}w_{22}\cdots w_{nn}$, wobei ${}_1, \ldots, {}_n \notin \Sigma$ neue Symbole sind.

b) 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.

c) Konstruiere nach Deiner Methode den Suffixbaum für die Menge

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

Aufgabe 3

Gegeben sei eine Menge $S = \{s_1, \ldots, s_\ell\}$ 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.

Aufgabe 4

a) Betrachten Sie einen Suffix-Baum. Sei $f()$ die Suffix-Link-Funktion und v ein Knoten im Suffixbaum. Zeigen Sie, dass v ist kein Blatt auch $f(v)$ ist kein Blatt impliziert.

b) Beweisen Sie, dass die Größe eine Suffixbaums für einen String der Länge n $\Theta(n^2)$ im Worst Case ist, falls die Kantenlabel explizit ausgeschrieben werden.