Sommersemester 2003
Dipl.-Inf. A. Thum, Dr. A. Gerisch

7. Übung „Algorithmen der Bioinformatik I“

Abgabe am 19. Juni 2003 in der Übung

Institut für Informatik
Martin-Luther-Universität Halle-Wittenberg

Aufgabe 1. Für die Implementierung eines Suffixbaums für einen String muß man sich eine geeignete Datenstruktur für die inneren Knoten überlegen. Ein Punkt dabei ist, wie die vom Knoten ausgehenden Kanten im Knoten repräsentiert werden können. Dabei spielen sowohl der benötigte Speicherbedarf als auch die Zugriffszeit auf eine gesuchte Kante (um sie zu verfolgen) eine wichtige Rolle.
Wenn Σ das Alphabet aller möglichen Zeichen im String ist, dann gibt es maximal $|\Sigma|$ ausgehende Kanten pro inneren Knoten. Damit können alle Kanten mit einem Feld der Länge $|\Sigma|$ repräsentiert werden (der erste Buchstabe der Kantenbeschriftung wird als Index verwendet). Alternativ können ausgehende Kanten über eine verkettete Liste oder einen balancierten binären Suchbaum im Knoten repräsentiert werden.
Vergleichen Sie die drei Möglichkeiten hinsichtlich ihres Zeitbedarfs zum Finden einer Kante und ihres Speicherbedarfs (ein Verweis auf ein Listenelement bzw. einen Kindknoten benötige genausoviel Speicher wie ein Verweis auf eine Kante). Überlegen Sie sich, in welchen Ebenen des Suffixbaums (obere, mittlere, untere) Sie welche Art der Repräsentation anwenden würden und begründen Sie Ihre Entscheidung.

Aufgabe 2. In einem generalisierten Suffixbaum für die Strings $S_1, S_2, ...S_m$ stehen in den Blättern Zahlenpaare (i, j), wobei (i, j) in einem Blatt bedeutet, dass die Pfadbeschriftung von der Wurzel bis zum Blatt gleich dem Suffix $S_i[j..]$ ist. Jeder Suffix jedes (im Baum enthaltenen) Strings definiert also ein Indexpaar und es gibt ein Blatt im Baum, das dieses Paar enthält. Ein Blatt kann mehrere Indexpaare enthalten, die aber alle von verschiedenen Strings stammen.
Wir wollen einen generalisierten Suffixbaum und für die zwei Strings $S_1 = xabxa\$$ und $S_2 = babxba\$$ bauen. Zeichen Sie den Suffixbaum für S_1 und fügen Sie den String S_2 mit dem inkrementellen Ukkonen-Algorithmus in den Baum ein. Zeichnen Sie nach dem Ende jeder Phase den Baum mit Indexpaaren in den Blattbeschriftungen und Kantenbeschriftungen als Teilstrings (keine komprimierte Darstellung über Indizes). Suffixlinks müssen nicht erzeugt werden.

Aufgabe 3. In einem generalisierten Suffixbaum sollen die Strings $S_1, S_2, ...S_m$ mit dem inkrementellen Ukkonen-Algorithmus abgelegt werden. Jeder dieser Strings wird dabei mit dem gleichen Endezeichen (\$) versehen.
Mit welcher Blattzahl b muss die erste Erweiterung beim Einfügen des Strings S_k mit $k > 1$ gestartet werden und was passiert in dieser Erweiterung (welche Erweiterungsregel, warum)?
Welche der Erweiterungsregeln I, IIa, IIb, III müssen wie geändert werden, damit nach dem Einfügen von S_k ein korrekter generalisierter Suffixbaum für $S_1, S_2, ..., S_k$ entsteht? *Hinweis:* Was passiert, wenn ein Suffix eines Strings auch Suffix eines anderen Strings ist?

Aufgabe 4. Wir betrachten eine Datenbank, in der $m \geq 1$ DNA-Sequenzen $S_1, S_2, ..., S_m$ abgespeichert sind. Nun soll eine weitere Sequenz S_{m+1} in die Datenbank eingefügt werden. Um die Datenbank vor redundanter Information zu bewahren, soll deshalb überprüft werden, ob S_{m+1} bereits Teilstring eines der in der Datenbank enthaltenen Strings ist oder ob es Teilstrings S_i in der Datenbank gibt, die in S_{m+1} enthalten sind.
Beschreiben Sie einen effizienten Algorithmus, der das geschilderte Problem löst. Die Laufzeit des Algorithmus soll proportional zur Gesamtlänge aller Strings $S_1, S_2, ..., S_{m+1}$ sein. Begründen Sie Ihre Laufzeitabschätzung.

Aufgabe 5. Diese Aufgabe beschäftigt sich mit dem Sequenzenalignment und Edit-Transkripten für zwei Strings S_1 und S_2. Zeigen Sie, dass es zu jedem Alignment von S_1 und S_2 einen Edit-Transkript gibt. Verwenden Sie dazu vollständige Induktion über die Länge des Alignment. Im Induktionsschritt müssen 3 Fälle unterschieden werden. Benennen Sie diese und beweisen Sie einen davon.