

Übungen zur Algorithmischen Bioinformatik I

Blatt 8

Abgabetermin: Montag, 26.6.2017, 10 s.t.

1. Aufgabe (Bonus-Aufgabe):

Konstruieren Sie einen Suffix-Baum für $t = t_1 \cdots t_{10} = aaabaababb$ mit Ukkonens Algorithmus aus der Vorlesung. Geben Sie dabei alle ausgeführten Zwischenschritte an. Markieren Sie dabei insbesondere die Position des aktiven Knotens und Endknotens im jeweiligen Suffix-Baum. Zeichnen Sie dabei die verwendeten und neu eingetragenen Suffix-Links mit jeweils einer anderen Farbe ein. Der Übersichtlichkeit wegen können als Kantenlabel die Zeichenreihen selbst anstelle der üblicherweise verwendeten Referenzen auf t verwendet werden.

2. Aufgabe (Bonus-Aufgabe):

Konstruieren Sie einen Algorithmus, der alle Vorkommen des Suchwortes $s \in \Sigma^m$ in einem Text $t \in \Sigma^n$ ausgibt und der auf einem Suffix-Baum für t basiert.

Geben Sie dazu den Algorithmus in Pseudo-Code an (nicht in Worten). Beweisen Sie die Korrektheit und analysieren Sie die Laufzeit. Folgende Operationen sollen dabei verwendet werden:

SuffixTree buildSuffixTree(String) Erzeugt einen Suffix-Baum.

Node getRoot(SuffixTree) Gibt die Wurzel des Suffix-Baums zurück.

Edge firstEdge(Node) Gibt die Kante zum ersten Kind zurück und NULL, wenn es keines gibt.

Edge nextEdge(Node, Edge) Gibt die Kante zum nächsten Kind zurück und NULL, wenn es kein weiteres gibt.

(int, int) getEdgeLabel(Edge) Gibt das Kanten-Label zurück.

Node getChild(Edge) Gibt den Kind-Knoten der Kante zurück.

3. Aufgabe:

Entwerfen Sie einen effizienten Algorithmus, der in einem Text t das längste Teilwort w findet, das genau zweimal in t als Teilwort auftritt.

Hinweis: Korrektheitsbeweis und Laufzeitanalyse nicht vergessen.