

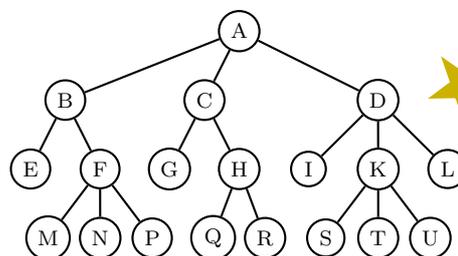
Algorithmen auf Sequenzen

Abgabetermin: Donnerstag, den 11. Januar vor der Vorlesung

Dieses Übungsblatt enthält eine zusätzliche Notenbonusaufgabe, wobei die erzielten Punkte als Bonus-Punkte gewertet werden, d.h. dass die erzielten Punkte für den Notenbonus berücksichtigt werden, die zu erzielenden Punkte jedoch nicht.

Aufgabe (Notenbonus) 1

Wende auf den rechten Baum die lineare Vorverarbeitung aus der Vorlesung an (mit $k = 5$) und beantworte die LCA-Anfragen $\text{lca}(N, S)$, $\text{lca}(E, N)$ und $\text{lca}(F, K)$ gemäß dem Algorithmus aus der Vorlesung. Dabei müssen die verwendeten Bit-Vektoren $V^{B,j}$ angegeben werden, aber nicht deren Herleitung.



Aufgabe (Notenbonus) 2

Zeige, wie man mit Hilfe des mit dem Vokabular von $t \in \Sigma^*$ markierten Suffix-Baumes $S(t\$)$ einen Algorithmus zur Ausgabe aller (rechts-verzweigenden) Tandem-Repeat-Paare entwickeln kann und analysiere seine Laufzeit. Welchen Vorteil hat dieser Algorithmus?

Aufgabe (Notenbonus) 3

Sei $t = w_1\# \cdots \#w_m \in (\Sigma \cup \{\#\})^*$ mit $w_i \in \Sigma^+$ und sei $n = |t|$.

Beschreibe einen Algorithmus, der t in Zeit $O(n)$ so vorverarbeitet, dass anschließend für jedes Wort p in Zeit $O(|p|)$ entschieden werden kann, ob p eines der w_i ist, wobei der zusätzlich zum Text t benötigte Platz der resultierenden Datenstruktur $O(m)$ sein soll (wobei die Zeichenreihe t selbst auch zur Verfügung steht).

Wie groß ist der (maximale) zusätzliche Platzbedarf der vorgeschlagenen Datenstruktur während der Konstruktion?

*Fröhliche Weihnachten und
einen guten Rutsch ins neue Jahr!*