Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Volker Heun

Wintersemester 2018/19
Übungsblatt 2
25. Oktober 2018

Algorithmen auf Sequenzen

Abgabetermin: Mittwoch, den 31. Oktober vor der Übung

Für den Notenbonus sind nur die entsprechend gekennzeichneten Aufgaben abzugeben. Die Aufgaben sind einzeln zu bearbeiten.

Bei einer elektronischen Abgabe sind alle Aufgaben als eine PDF-Datei zu versenden (an `Sophie.Friedl@bio.ifi.lmu.de`). Der Dateiname muss Vor- und Nachname sowie die Nummer des Übungsblatts enthalten.

Aufgabe (Notenbonus) 1

Sei $S > 0$ der Score einer *maximal scoring subsequence* einer Folge $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$. Gib einen möglichst effizienten Algorithmus (in Pseudo-Code) an, der *alle* Teilfolgen mit Score S findet. Der Wert S ist natürlich a priori nicht bekannt.

Analysiere den Algorithmus hinsichtlich Zeit- und Platzkomplexität.

Hinweis: Wenn für $i \leq k \leq \ell \leq j \in [1 : n]$ mit $(i < k)$ oder $(\ell < j)$ gilt, dass $\sigma(i, j) = \sigma(k, \ell) = S$, dann stellt $(a_i, \ldots, a_j)$ auch hier keine Lösung dar und ist nicht auszugeben.

Aufgabe (Notenbonus) 2

Entwirf einen Linearzeit-Algorithmus für MAltSS und analysiere ihn.

Hinweis: Korrektheitsbeweis und Laufzeitanalyse nicht vergessen!

<u>Maximal Alternating Scoring Subsequence (MAltSS)</u>

Eingabe: Eine Folge $(a_1, \ldots, a_n) \in \mathbb{R}^n$.
Ausgabe: Eine Teilfolge $(a_i, \ldots, a_j)$ mit $i \leq j \in [1 : n]$, die den Wert $\alpha(i, j)$ maximiert, wobei $\alpha(i, j) = \sum_{\ell=i}^{j} (-1)^{\ell-i} \cdot a_\ell$.

Aufgabe 3

Beweise, dass jeder gewurzelte Baum, der keinen Knoten mit genau einem Kind besitzt (mit Ausnahme der Wurzel), höchstens so viele innere Knoten wie Blätter besitzt.