
Algorithmen auf Sequenzen

Abgabetermin: Samstag, den 2. November, 10⁰⁰ in Moodle

Aufgabe 1

Sei $a = (a_1, \dots, a_n) \in \mathbb{R}^n$ eine Folge reeller Zahlen und sei $S > 0$ der Score einer *maximal scoring subsequence* von a . Gib einen möglichst effizienten Algorithmus an, der *alle* Teilfolgen von a mit Score S findet. Der Wert S ist natürlich a priori nicht bekannt und der Algorithmus zu AMSS darf nicht verwendet werden.

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, \dots, a_j) auch hier keine Lösung dar und ist nicht auszugeben.

Aufgabe 2

Sei $a = (a_1, \dots, a_n) \in \mathbb{R}^n$ eine Folge sowie $a' = (a_{i_1}, \dots, a_{j_1})$ und $a'' = (a_{i_2}, \dots, a_{j_2})$ zwei Teilfolgen von a mit maximalem Score. Es soll dabei angenommen werden, dass wenn a'' eine echte Teilfolge von a' ist (d.h. $i_1 \leq i_2 \leq j_2 \leq j_1$), nun nur die längere Teilfolge a' als Lösung gewertet werden soll, aber nicht a'' .

Zeige, dass es auch unter dieser Annahme keine echt überlappenden Lösungen geben kann.