Übungen zur Vorlesung Sequenzanalyse

Universität Bielefeld, SS 2017
Prof. Dr. Jens Stoye · M.Sc. Tizian Schulz

https://gi.cebitec.uni-bielefeld.de/teaching/2017summer/sa

Übungsblatt 1 vom 20.04.2017
Abgabe am 02.05.2017 vor Beginn der Vorlesung

Organisatorisches

Wichtige Hinweise: Bitte unbedingt im eKVV für die Vorlesung und die Übungen (Sammeltermin und dem jeweiligen einzelnen Übungstermin) registrieren. Auf der Homepage der Veranstaltung steht eine elektronische Version des Skripts zum Download bereit. Die gedruckte Version wird in der ersten Übung verteilt.

Abgabe der Übungszettel: Die Bearbeitung der Aufgaben darf in Gruppen (maximal zwei Personen) erfolgen. Sie können entweder **vor** Beginn der Vorlesung am Dienstag im Hörsaal, im Briefkasten vor U10-151 oder in elektronischer Form als E-Mail an den jeweiligen Tutor abgegeben werden. Alternativ könnt ihr die Zettel auch euren Tutoren persönlich geben. Bitte den Tutorientermin bzw. den Tutor, sowie euren Namen und den eures Gruppenpartners deutlich auf der Abgabe vermerken.

Quellenangabe auf den Übungszetteln: Benutzt ihr bei der Bearbeitung der Übungsaufgaben andere Quellen als das Skript zur Vorlesung, so gebt diese bitte immer mit an. Ist nicht ersichtlich, woher ihr eure Informationen genommen habt, können wir euch leider keine Punkte geben.

Teilnahme an der Klausur: Die Klausur kann am Ende mitschreiben, wer mindestens 50% der Übungspunkte erreicht und mindestens zweimal in den Übungen eine Aufgabe vorgerechnet hat.

Aufgaben

Aufgabe 1 (Komplexitätsklassen) (4 Punkte)

Gegeben sind zwei Algorithmen $A1$ und $A2$. Bei einer Eingabegröße von n braucht $A1$ $f_1 = \frac{n^2}{8} + \frac{n}{4}$ Rechenschritte, $A2$ benötigt $f_2 = 15n \cdot \log n$ Schritte.

1. Stelle die Funktionen grafisch in einem gemeinsamen Koordinatensystem dar. Es reicht, wenn du die Funktionen im Bereich $n = 0$ bis $n = 5$ skizzierst.
2. Ermittle die Komplexitätsklasse der Algorithmen in der $\mathcal{O}$-Notation.
3. Welcher Algorithmus hat die asymptotisch schlechtere Laufzeit? Ab welchem n_0 wird das deutlich, mit der Vorgabe, dass $c = 1$ ist?

Aufgabe 2 (Substrings) (4 Punkte)

Gegeben sei das Wort $s = \texttt{NILPFERD}$.

1. Gib alle Substrings der Länge 3 und eine Subsequenz der Länge 5 von s an.
2. Gib alle Präfixe und Suffixe von s an.
3. Versuche, eine allgemeine Formel für die Anzahl der Substrings einer festen Länge k in einem beliebigen Wort der Länge n zu finden. Wie viele Substrings der Länge 4 enthält s?

Aufgabe 3 (Diskrete Metrik) (3 Punkte)

Auf jeder Menge $\mathcal{X}$ lässt sich die diskrete Metrik $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ definieren:

$$d(x, y) := \begin{cases} 0, & \text{wenn } x = y, \\ 1, & \text{wenn } x \neq y. \end{cases}$$

Weise nach, dass es sich bei d um eine Metrik auf $\mathcal{X}$ handelt, indem Du die Definitionen (3.1) - (3.3) aus dem Skript (S. 14) überprüfst.

Aufgabe 4 (Implementierung der Hamming-Distanz) (4 Punkte)

In der Vorlesung wurde die Hamming-Distanz eingeführt. Implementiere eine Funktion, die als Parameter zwei Strings a und b übergeben bekommt und ihre Hamming-Distanz ausgibt. Wenn die beiden Strings eine unterschiedliche Länge haben, soll eine Fehlermeldung ausgegeben werden.

Benutze hierfür die Programmiersprache Haskell.