Stefan Kurtz **Universität Hamburg**
Abteilung für Genominformatik **Zentrum für Bioinformatik**

Grundlagen der Sequenzanalyse
Wintersemester 2022/2023
Übungen zur Vorlesung: Ausgabe am 01.11.2022

In den Aufgaben sind die Namen der Dateien, in denen Sie Ihre Lösungen beschreiben sollen, vorgegeben. Sie können Ihre Lösungen entweder als Textdatei (Endung .txt) oder in LaTeX (Endung .tex) formatieren. Falls Sie LaTeX verwenden, bitte nur die .tex-Datei im Verzeichnis hinterlassen.

Aufgabe 2.1 (2 Punkte) Verifizieren Sie bitte jeweils, ob die folgenden Aussagen korrekt sind oder nicht.

1. Die Funktion f_1 mit $f_1(n) = 2^{n+1}$ ist in $O(2^n)$.
2. Die Funktion f_2 mit $f_2(n) = 2^{2n}$ ist in $O(2^n)$.

Begründen Sie Ihre Antwort. Schreiben Sie Ihre Lösung in die Datei `verification.tex`.

Aufgabe 2.2 (4 Punkte) Sei $\mathcal{A} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}, \mathtt{N}\}$ das erweiterte DNA-Alphabet. Dabei steht `N` für eine sog. *Wildcard*, d.h. eine unbestimmte Base. Sei $s \in \mathcal{A}^n$ eine DNA-Sequenz der Länge n.

Sei $k > 0$. Ein k-mer in s ist ein Substring von s der Länge k, der keine Wildcard enthält. Beispiel: In der folgenden Tabelle sind für verschiedene Strings und Werte von k die Startpositionen der k-mere angegeben:

s	k	Startpositionen der k-mere
ACGNTGANNNAAANATATNA	4	15
AAATTNAAA	5	1
AAANATAG	3	1, 5, 6

Geben Sie Pseudocode für einen Algorithmus an, der in einer Laufzeit von $O(n)$ für ein gegebenes k mit $1 \leq k \leq n$ die Positionen aller k-mere in einer Sequenz s der Länge n ausgibt. Die Laufzeit darf also nicht von k abhängen. Ein Algoritmus, der für jeden Substring von s der Länge k prüft, ob der Substring eine Wildcard enthält, hat die Laufzeit $O(kn)$ und erfüllt damit die Anforderungen nicht. Der Algorithmus darf also für jede Position von s nur eine konstante Anzahl von Berechnungsschritten durchführen.

Hinweis: Um die lineare Laufzeit zu erreichen ist es hilfreich, relevante Information aus den bisherigen Schritten des Algorithmus zu speichern, so dass nicht immer alles neu berechnet werden muss.

Schreiben Sie Ihre Lösung in die Datei `kwords_without_wildcards.tex`.

Aufgabe 2.3 (6 Punkte) Sei w ein nichtleerer String. Eine Randsequenz von w ist ein echter Substring von w, der sowohl Präfix von w als auch Suffix von w ist. Beispiel: Die Randsequenzen von $w =$ `abaababaaba` sind ε, `a`, `aba` und `abaaba`. Randsequenzen spielen in verschiedenen Algorithmen zur effizienten Suche von Mustern in Sequenzen eine wichtige Rolle. Diese Aufgabe besteht aus vier Teilen:

1. Berechnen Sie für die 8 Strings in Zeile 2-9 aus der Datei `rand_sequenzen.tsv` (siehe Material) jeweils alle nichtleeren Randsequenzen und schreiben Sie diese mit absteigender Länge nach dem String in die gleiche Zeile dieser Datei, jeweils mit einem Tabulator getrennt. Dieses Format erleichtert die Korrektur Ihrer Lösung. Die erste Zeile der Datei enthält bereits den String aus dem obigen Beispiel und dessen Randsequenzen. 4 Pkt
2. Beschreiben Sie in einer Datei `rand_sequenzen_eigenschaft.txt`, in welcher nichttrivialen Beziehung zwei aufeinanderfolgende Randsequenzen des gleichen Strings stehen. D.h. welche Eigenschaft hat eine Randsequenz bzgl. ihres Vorgängers? Eine triviale Eigenschaft wäre, dass die eine Randsequenz eine geringere Länge hat, als die andere. Ein oder zwei Sätze sollten reichen, um diese Beziehung/Eigenschaft zu beschreiben. Es ist nicht erforderlich, diese Eigenschaft allgemein zu beweisen. Bei einem Blick auf die Beispiele sollte die Eigenschaft gut erkennbar sein. 0.5 Pkt
3. Die Randsequenzen-Tabelle für einen String w der Länge n ist eine Tabelle rst mit n nichtnegativen ganzen Zahlen, so dass für alle i, $1 \leq i \leq n$ gilt: $rst[i]$ ist die Länge der längsten Randsequenz von $w[1 \ldots i]$.

   Beispiel: Sei $w =$ `abaababaaba`. Dann sind 0, 0, 1, 1 die ersten 4 Werte der Randsequenzen-Tabelle von w.

   Geben Sie in der letzten Zeile der Datei `rand_sequenzen.tsv` die Randsequenzen-Tabelle von $w =$ `abaababaaba` an und zwar als Folge von durch Tabulatoren getrennten Werten. 1 Pkt
4. Beschreiben Sie in einer Datei `rand_sequenzen_max.txt`, wie man aus der Randsequenzen-Tabelle eines Strings w die längste Randsequenz von w bestimmen kann. Hier reicht eine kurze textuelle Beschreibung. 0.5 Pkt

Bitte die Lösungen zu diesen Aufgaben bis zum 06.11.2022 um 22:00 Uhr an gsa@zbh.uni-hamburg.de schicken.