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

Sommersemester 2022
Semestralklausur
04. August 2022

Algorithmische Bioinformatik I

Vorname	Name	Matrikelnummer

Reihe	Platz	Unterschrift

Allgemeine Hinweise zur Semestralklausur

- Vor der Prüfung ist diese Seite mit Vornamen, Namen, Matrikelnummer, Reihe und Platz **leserlich mit Druckbuchstaben** zu versehen und zu unterschreiben.
- Bitte **nicht** in *roter* oder *grüner* Farbe bzw. **nicht** mit Bleistift schreiben.
- Der Studentenausweis und ein amtlicher Lichtbildausweis sind bereit zu halten.
- Die reine Bearbeitungszeit beträgt **120 Minuten**.
- Es sind insgesamt 40 Punkte zu erreichen, zum Bestehen sind 17 Punkte nötig.

Viel Erfolg!

Hörsaal verlassen von .... bis .... von .... bis ....

Vorzeitig abgegeben um ....

	Hz	A1	A2	A3	A4	A5	$\sum$
Erstkorrektur							
Nachkorrektur							
Zweitprüfer							

Vorname: ______________ Name: ______________ Matrikelnummer: ____________

Aufgabe 1 (8 Punkte)

Gib das Master-Theorem aus der **Vorlesung** an. Spezifiziere hierzu insbesondere die drei verschiedenen Fälle und gib an, welche Lösung der jeweilige Fall besitzt.

Bestimme die Asymptotik von $T(n)$ mithilfe des Master-Theorems aus der **Vorlesung** unter Angabe einer der drei Fälle (siehe oben) mit Begründung bzw. begründe, warum das Master-Theorem nicht anwendbar ist. Es gilt dabei immer $T(1) = 1$:

a) $T(n) = 4 \cdot T(n/3) + n^2$.

b) $T(n) = 2 \cdot T(n/2) + \sqrt{n}$.

c) $T(n) = 9 \cdot T(n/3) + n^2$.

Lösungsskizze

Seien $a, b, d \in \mathbb{N}$ mit $b > 1$, sei $f(n)$ eine Funktion und sei $\mathcal{C}(n)$ definiert durch die Rekursionsgleichung $\mathcal{C}(n) = a \cdot \mathcal{C}(n/b) + f(n)$ für $n > 1$ und $\mathcal{C}(1) = d$. Dann gilt:

$$\mathcal{C}(n) = \begin{cases} \Theta(n^{\log_b(a)}) & \text{falls } f(n) = O(n^{\log_b(a)-e}) \text{ für ein konstantes } e > 0 \\ \Theta(n^{\log_b(a)}\log(n)) & \text{falls } f(n) = \Theta(n^{\log_b(a)}) \\ \Theta(f(n)) & \text{falls } f(n) = \Omega(n^{\log_b(a)+e}) \text{ für ein konstantes } e > 0 \\ & \text{und } a \cdot f(n/b) \leq c \cdot f(n) \text{ für ein konstantes } c < 1 \end{cases}$$

a) Für das Master-Theorem erhalten wir $a = 4, b = 3$ und $f(n) = n^2$. Es gilt $\log_3(4) \in (1, 2)$ und somit $f(n) = n^2 = \Omega(n^{\log_3(4)+e}) = \Omega(n^{\log_b(a)+e})$. für ein geeignetes $e \in (0, 2-\log_3(4))$. Weiter ist
$$a \cdot f(n/b) = 4 \cdot (n/3)^2 \leq \frac{4}{9} \cdot n^2 \overset{!}{\leq} c \cdot f(n)$$
und somit gilt der dritte Fall des Master-Theorems mit $c = 4/9 < 1$. Damit gilt $T(n) = \Theta(f(n)) = \Theta(n^2)$.

b) Für das Master-Theorem erhalten wir $a = 2$, $b = 2$ und $f(n) = \sqrt{n}$. Es gilt $\log_2(2) = 1$ und somit $f(n) = \sqrt{n} = n^{1/2} = O(n^{1-e}) = O(n^{\log_2(2)-e}) = O(n^{\log_b(a)-e})$ für ein geeignetes $e \in (0, 1/2)$. Somit gilt der erste Fall des Master-Theorems und damit dann $T(n) = \Theta(n^{b\log(a)}) = \Theta(n^{\log_2(2)}) = \Theta(n)$.

c) Für das Master-Theorem erhalten wir $a = 9$, $b = 3$ und $f(n) = n^2$. Es gilt $\log_9(3) = 2$ und somit $f(n) = n^2 = \Theta(n^{\log_3(9)}) = \Theta(n^{\log_b(a)})$. Also gilt der zweite Fall des Master-Theorems und es ist $T(n) = \Theta(n^{\log_3(9)}\log(n)) = \Theta(n^2\log(n))$.

Vorname: __________ Name: __________ Matrikelnummer: __________

Aufgabe 2 (8 Punkte)

Betrachte den unter a) abgebildeten Suffix-Baum für $s = s_1 \cdots s_6 = abaaba$. Der besseren Lesbarkeit wegen sind hierbei immer explizit die Kantenlabels statt der Referenzen angegeben.

a) Zeichne alle Suffix-Links ein, die Ukkonens Algorithmus hierfür konstruiert hat.

b) Gib die Kantenlabels so an, wie sie in Ukkonens Algorithmus verwendet werden.

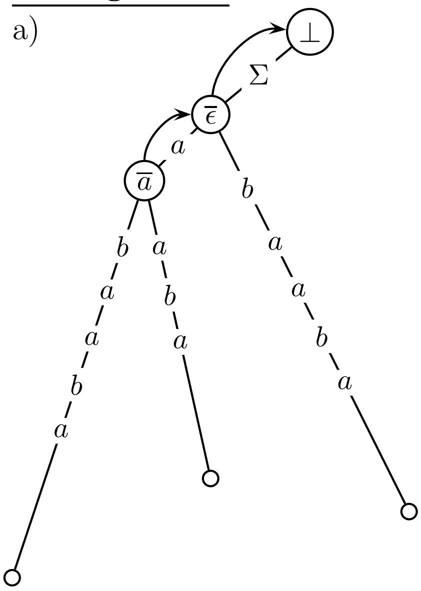
c) Führe Ukkonens Algorithmus für den Übergang von s auf $s' = s \cdot a = abaabaa$ aus.

d) Führe Ukkonens Algorithmus für den Übergang von s' auf $s'' = s' \cdot a = abaabaaa$ aus.

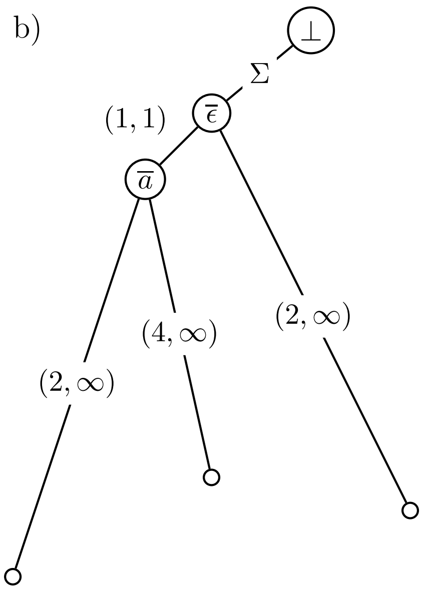
Gib für c) und d) alle Zwischenschritte an, markiere insbesondere die Position des aktiven Knotens und Endknotens im jeweiligen Suffix-Baum. Zeichne dabei nur die verwendeten und neu eingetragenen Suffix-Links mit jeweils einer anderen Farbe ein und nummeriere die neuen Blätter in der Reihenfolge der Einfügung.

Lösungsskizze

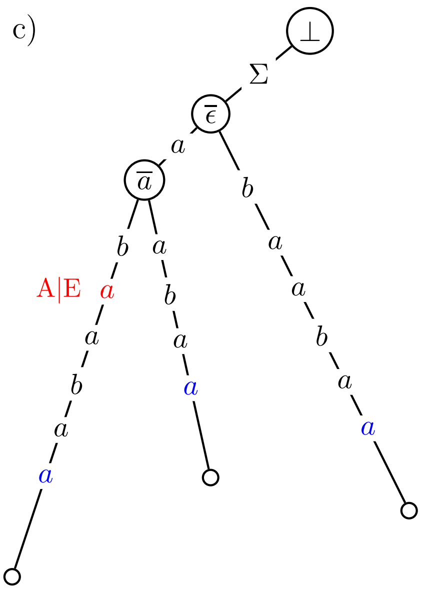
a)



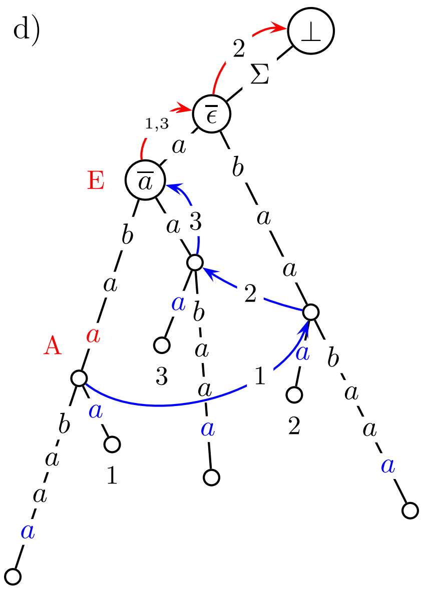
b)



c)



d)



Aufgabe 3 (8 Punkte)

Betrachte die Wörter s = CORONA und t = SCORES. Berechne den ersten Schritt des Hirschberg-Algorithmus bei einem globalen Sequenzen-Alignment für s und t zur Rekonstruktion des Tracebacks. Bestimme insbesondere den bzw. die Schnittpunkte der Wörter s und t, d.h. die Teilwörter, für die der Hirschberg-Algorithmus rekursiv aufgerufen wird und gib das bzw. die zugehörigen Alignments an.

Die beiden rekursiven Aufrufe müssen nicht ausgeführt werden, das jeweilige Alignment kann von Hand ermittelt werden. Zeichne dabei in der Tabelle auch die Traceback-Pfeile ein (die vom Hirschberg-Algorithmus nicht verwendet werden).

Die Kostenfunktion für ein Distanzmaß sei dabei mit 0 für ein Match, mit 3 für eine Substitution und mit 2 für eine Indel-Operation gegeben.

Lösungsskizze

		S	C	O	R	E	S
	0	2	4	6	8	10	12
C	2	3	2	4	6	8	10
O	4	5	4	2	4	6	8
R	6	7	6	4	2	4	6
	=18	=17	=14	=13	=10	=11	=12
	12	10	8	9	8	7	6
O	14	12	10	8	6	5	4
N	13	11	9	7	5	3	2
A	12	10	8	6	4	2	0

Damit liegt der Schnittpunkt bei $(3, 4)$, d.h. $COR|ONA$ versus $SCOR|ES$.

Die möglichen Alignments lauten:

$$\begin{pmatrix} -COR|ONA \\ SCOR|ES- \end{pmatrix}, \quad \begin{pmatrix} -COR|ONA \\ SCOR|E-S \end{pmatrix}, \quad \begin{pmatrix} -COR|ONA \\ SCOR|-ES \end{pmatrix}.$$

Vorname: ______________ Name: ______________ Matrikelnummer: ______________

Aufgabe 4 (8 Punkte)

Beweise $\Delta(n \cdot H_{n-1}) = H_n + 1$ und berechne damit sowie mithilfe der diskreten Stammfunktion und partieller Integration für $n \in \mathbb{N}$:

$$\sum_{i=1}^{n-1} H_i \cdot (H_i + 1)$$

Erinnerung: Es gilt $H_n = \sum_{j=1}^{n} \frac{1}{j}$.

<u>Lösungsskizze</u>

Es gilt nach Definition und der Produktregel:

$$\begin{aligned}
\Delta(n \cdot H_{n-1}) &= (\Delta(n)) \cdot (H_n) + n \cdot (\Delta(H_{n-1})) \\
&= H_n + n \cdot (H_n - H_{n-1}) \\
&= H_n + n \cdot \left(\sum_{j=1}^{n} \frac{1}{j} - \sum_{j=1}^{n-1} \frac{1}{j} \right) \\
&= H_n + n \cdot \frac{1}{n} \\
&= H_n + 1
\end{aligned}$$

Nach der partiellen Integration gilt $\sum (f \cdot \Delta g) = fg - \sum (\Delta f)(Eg)$ mit $f(x) = H_x$ und $g(x) = x \cdot H_{x-1}$ und daher $(\Delta g)(x) = H_{x-1} + 1$ sowie $(\Delta f)(x) = x^{\underline{-1}}$:

$$\begin{aligned}
\sum_{i=1}^{n-1} H_i \cdot (H_i + 1) &= [H_i \cdot (i \cdot H_{i-1})]_1^n - \sum_{i=1}^{n-1} i^{\underline{-1}} \cdot (i+1) H_i \\
&= n \cdot H_n \cdot H_{n-1} - 0 - \sum_{i=1}^{n-1} \frac{1}{i+1} \cdot (i+1) H_i \\
&= n \cdot H_n \cdot H_{n-1} - \sum_{i=1}^{n-1} (H_i + 1 - 1) \\
&= n \cdot H_n \cdot H_{n-1} - \sum_{i=1}^{n-1} (H_i + 1) + (n-1) \\
&= n \cdot H_n \cdot H_{n-1} - [i \cdot H_{i-1}]_1^n + (n-1) \\
&= n \cdot H_n \cdot H_{n-1} - [n \cdot H_{n-1} - 0] + (n-1) \\
&= n \cdot H_n \cdot H_{n-1} - n \cdot H_{n-1} + n - 1
\end{aligned}$$

Vorname: ______________ Name: ______________ Matrikelnummer: ______________

Aufgabe 5 (8 Punkte)

Für ein Wort $w = w_1 \cdots w_n \in \Sigma^*$ ist seine *Spiegelung* definiert als $w^R := w_n \cdots w_1 \in \Sigma^*$. Ein Wort $w' \in \Sigma^*$ ist eine *Doppelspiegelung* eines Wortes $w \in \Sigma^*$, wenn es zwei Worte $x, y \in \Sigma^*$ mit $w = xy$ gibt, so dass $w' = x^R y^R$gilt.

Beispielsweise ist LAGERTOR eine Doppelspiegelung von REGALROT.

Konstruiere einen Algorithmus, der für zwei Zeichenketten s und t in linearer Zeit ermittelt, ob s eine Doppelspiegelung von t ist.

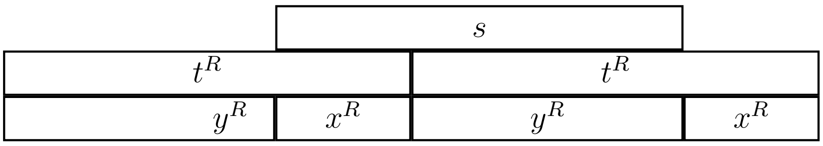
Hinweis: Korrektheitsbeweis und Laufzeitanalyse nicht vergessen!

Lösungsskizze

Sei $s, t \in \Sigma^n$. Ist $|s| \neq |t|$, dann kann s nach Definition keine Doppelrotation sein.

Wir suchen mithilfe des Algorithmus von Knuth, Morris und Pratt das Wort s in $t^R t^R$. Dies hat einen Zeitbedarf von $O(|s| + |t^R t^R|) = O(|s| + 2|t|) = O(|s| + |t|) = O(n)$, ist also linear in der Eingabegröße $|s| + |t| = \Theta(n)$.

Finden wir s in $t^R t^R$, dann gilt Folgendes, wenn wir $t = xy$ schreiben, wobei sich x und y (sowohl x als auch y können leer sein) als Teilwörter von t in Abhängigkeit des Auftretens von s in $t^R t^R$ ergeben:



Also ist $s = x^R y^R$ eine Doppelspiegelung von w.

Auf der anderen Seite sei $s = x^R y^R$ eine Doppelspiegelung von $t = xy$ für geeignete $x, y \in \Sigma^*$. Dann ist s offensichtlich ein Teilwort von $t^R t^R = y^R x^R y^R x^R$.

Alternative: Wir berechnen die Z-Boxen für $t'_0 \cdots t'_{|s|+2|t|} = s\sharp t^R t^R$. Gemäß der obigen Überlegung und Skizze ist genau dann s eine Doppelspiegelung von t, wenn es einen Index $i \in [|s| + 1 : |s| + |t|]$ gibt, so dass $Z_i \geq |s|$ für t' ist.

Die Z-Boxen lassen sich in Zeit $O(|s\sharp t^R t^R|) = O(n)$ berechnen. Die Suche nach dem Z-Wert ist ebenfalls in Zeit $O(n)$ möglich.