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

Sommersemester 2019
Wiederholungsklausur
08. Oktober 2019

Algorithmische Bioinformatik I

Vorname	Name	Matrikelnummer

Reihe	Platz	Unterschrift

Hiermit stimme ich einer Veröffentlichung meines Klausurergebnisses dieser Semestralklausur unter Verwendung meiner Matrikelnummer im Internet zu. *Ja* ☐ *Nein* ☐

____
(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) $T(n) = 8 \cdot T(n/4) + n\sqrt{n}$.

b) $T(n) = 3 \cdot T(n/4) + n\log(n)$.

c) $T(n) = 4 \cdot T(n/2) + n^2/\log(n)$.

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 $a = 8$, $b = 4$ und $f(n) = \sqrt{n}$. Es gilt dann $\log_4(8) = \frac{\log_2(8)}{\log_2(4)} = \frac{3}{2}$ und somit $f(n) = n\sqrt{n} = n^{3/2} = \Theta(n^{\log_4(8)}) = \Theta(n^{\log_b(a)})$. Also gilt der zweite Fall des Master-Theorems und $T(n) = \Theta(f(n)\log(n)) = \Theta(n\sqrt{n}\log(n))$.

b) Für das Master-Theorem erhalten wir $a = 3$, $b = 4$ und $f(n) = n\log(n)$. Es gilt $\log_4(3) \in (0, 1)$ und somit $f(n) = n\log(n) = \Omega(n^{\log_4(3)+e}) = \Omega(n^{\log_b(a)+e})$ für ein geeignetes $e \in (0, 1 - \log_4(3))$. Weiter ist

$$a \cdot f(n/b) = 3 \cdot \frac{n}{4}\log\left(\frac{n}{4}\right) \leq \frac{3}{4} \cdot n\log(n) \overset{!}{\leq} c \cdot f(n)$$

und somit gilt der dritte Fall des Master-Theorems mit $c = \frac{3}{4} < 1$. Damit gilt $T(n) = \Theta(f(n)) = \Theta(n\log(n))$.

c) Für das Master-Theorem erhalten wir $a = 4$, $b = 2$ und $f(n) = n^2/\log(n)$. Somit ist $\log_b(a) = \log_2(4) = 2$ und $f(n) = o(n^2)$. Es gilt weder

$$f(n) = n^2/\log(n) = \Omega(n^{2+e}) = O(n^{\log_b(a)+e})$$

für ein $e > 0$ noch

$$f(n) = n^2/\log(n) = \Theta(n^2) = \Theta(n^{\log_b(a)})$$

noch

$$f(n) = n^2/\log(n) = O(n^{2-e}) = O(n^{\log_b(a)-e})$$

für ein $e > 0$, da $\log(n) = o(n^e)$ für alle $e > 0$ und somit $\frac{1}{\log(n)} = \omega(n^{-e})$.

Vorname: ____________ Name: ____________ Matrikelnummer: ____________

Aufgabe 2 (8 Punkte)

Betrachte den unter a) abgebildeten Suffix-Baum für $s = s_1 \cdots s_7 = abbabba$. 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.

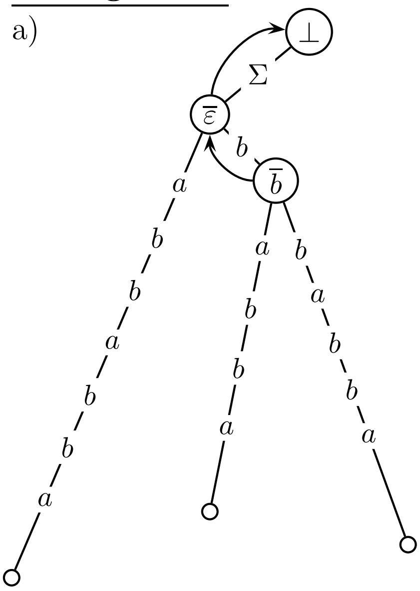
c) Führe Ukkonens Algorithmus für den Übergang von s auf $s' = s{\cdot}b = abbabbab$ aus.

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

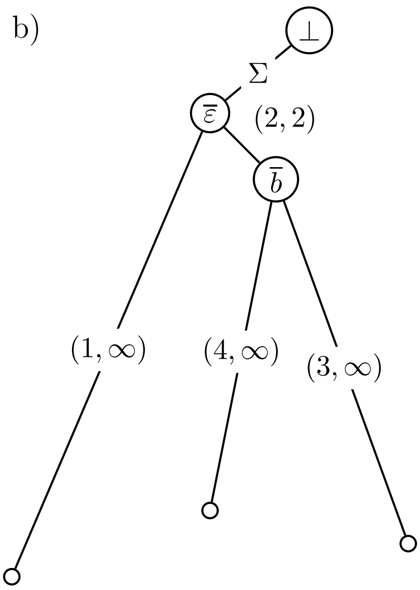
Gib für c) und d) alle Zwischenschritte an, markiere insbesondere die Position des aktiven Knotens und Endknotens im jeweiligen Suffix-Baum. Zeichne dabei nur die verwendeten und neu eingetragenen Suffix-Links mit jeweils einer anderen Farbe ein und nummeriere die neuen Blätter in der Reihenfolge der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

Lösungsskizze

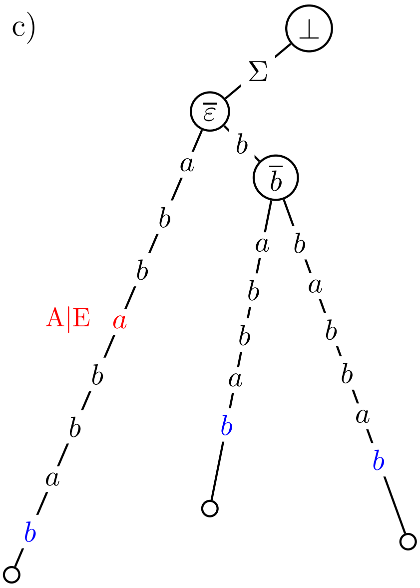
a)



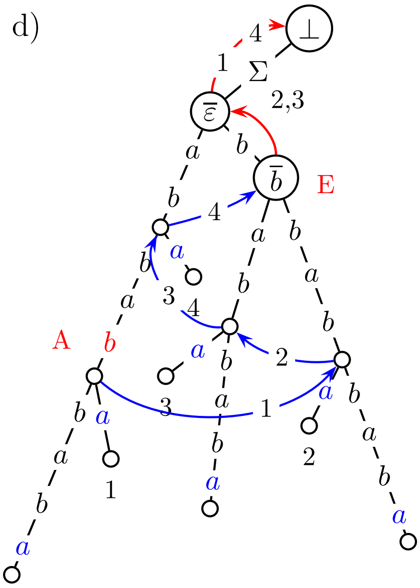
b)



c)



d)



Vorname: ______________ Name: ______________ Matrikelnummer: ______________

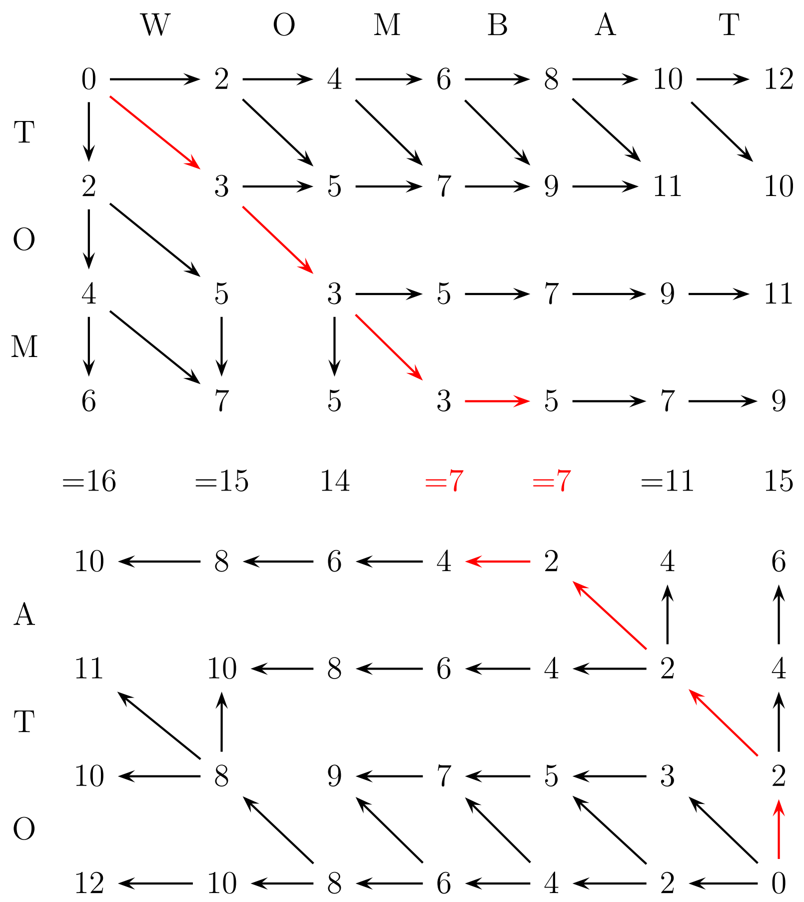
Aufgabe 3 (8 Punkte)

Betrachte die Wörter s = TOMATO und t = WOMBAT. 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.

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



Damit liegt der Schnittpunkt bei $(3, 3)$, d.h. $TOM|ATO$ versus $WOM|BAT$, oder $(3, 4)$, d.h. $TOM|ATO$ versus $WOMB|AT$.

Die beiden Alignments lauten gleich:

$$\begin{pmatrix} TOM{-}ATO \\ WOMBAT{-} \end{pmatrix}.$$

Vorname: ______________ Name: ______________ Matrikelnummer: ____________

Aufgabe 4 (8 Punkte)

Löse die folgende Rekursionsgleichung **mit Hilfe von erzeugenden Funktionen**:

$$f_n = 3 \cdot f_{n-1} + 1 \quad \text{für } n \geq 1, \qquad \text{und} \qquad f_0 = 1.$$

Erinnerung: Für $a \in \mathbb{R}$ mit $|z| < |1/a|$ gilt $\quad \dfrac{1}{1-az} = \sum_{n=0}^{\infty} (az)^n.$

Für zwei Folgen a_n und b_n gilt das Cauchy-Produkt: $\quad \sum_{n=0}^{\infty} a_n \cdot \sum_{n=0}^{\infty} b_n = \sum_{n=0}^{\infty} \sum_{k=0}^{n} a_k \cdot b_{n-k}.$

Lösungsskizze

Es gilt:

$$\begin{aligned}
F(z) &= \sum_{n=0}^{\infty} f_n z^n \\
&= f_0 + \sum_{n=1}^{\infty} f_n z^n \\
&= 1 + \sum_{n=1}^{\infty} (3 \cdot f_{n-1} + 1) z^n \\
&= 3 \sum_{n=0}^{\infty} f_n z^{n+1} + 1 + \sum_{n=1}^{\infty} z^n \\
&= 3z \sum_{n=0}^{\infty} f_n z^n + \sum_{n=0}^{\infty} z^n \\
&= 3z \cdot F(z) + \sum_{n=0}^{\infty} z^n.
\end{aligned}$$

Also gilt $F(z)\,[1-3z] = \sum_{n=0}^{\infty} z^n$ und somit für $|z| < 1$:

$$\begin{aligned}
F(z) &= \frac{1}{1-3z} \cdot \sum_{n=0}^{\infty} z^n \\
&= \sum_{n=0}^{\infty} (3z)^n \cdot \sum_{n=0}^{\infty} z^n \\
&\quad \text{mittels Cauchy-Produkt} \\
&= \sum_{n=0}^{\infty} \sum_{i=0}^{n} (3z)^i \cdot z^{n-i} \\
&= \sum_{n=0}^{\infty} z^n \sum_{i=0}^{n} 3^i \\
&= \sum_{n=0}^{\infty} \frac{3^{n+1}-1}{3-1} \cdot z^n \\
&= \sum_{n=0}^{\infty} \frac{(3^{n+1}-1)}{2} z^n
\end{aligned}$$

Also ist $f_n = \frac{3^{n+1}-1}{2}$.

Probe: $\frac{3^{0+1}-1}{2} = \frac{2}{2} = 1 = f_0$ und $3 \cdot f_{n-1} + 1 = 3 \cdot \frac{3^{(n-1)+1}-1}{2} + 1 = \frac{3^{n+1}-3}{2} + \frac{2}{2} = \frac{3^{n+1}-1}{2} = f_n$.

Aufgabe 5 (8 Punkte)

Sei Σ ein Alphabet, seien $a, b \in \Sigma^*$ und sei w ein Kostenmaß $w : \overline{\Sigma}_0^2 \to \mathbb{R}_+$ mit $w(x, x) = 0$ und $w(x, y) = w(x, -) = w(-, y) = 1$ für $x \neq y \in \Sigma$.

Wenn in einem Alignment $(\overline{a}, \overline{b}) \in \mathcal{A}(a, b)$ zwei Substitutionen (x, y) und (y, x) mit $x \neq y \in \Sigma$ unmittelbar hintereinander vorkommen, wird dies eine *Transposition* genannt. Solche Transpositionen werden nun mit Kosten 1 anstelle von zwei Substitutionen mit insgesamt Kosten 2 bewertet.

Für ein Alignment $(\overline{a}, \overline{b}) \in \mathcal{A}(a, b)$ bezeichnet w^+ die zugehörigen *erweiterten Kosten*. Dies sind die minimalen Kosten, die unter Berücksichtigung von Transpositionen für ein Alignment möglich sind. Allgemein sind die Kosten für Insertionen, Deletionen, Substitutionen und Transpositionen dann jeweils 1.

Beispiele: $w^+\binom{Konten}{Knoten} = 1$ und $w^+\begin{pmatrix} xyx \\ yxy \end{pmatrix} = 2$. Bei letzterem handelt es sich um eine Transposition (xy, yx) und eine Substitution (x, y) oder um eine Substitution (x, y) und eine Transposition (yx, xy).

Die zur erweiterten Kostenfunktion w^+ gehörige *erweiterte Alignment-Distanz* wird mit $\overline{d}^+$ bezeichnet.

Beispiel: $\overline{d}^+(xxyx, xyxx) = 1$.

Finde einen möglichst effizienten Algorithmus (mit Laufzeit $O(nm)$), der die erweiterte Alignment-Distanz für zwei Wörter $a \in \Sigma^m$ und $b \in \Sigma^n$ findet.

Hinweis: Korrektheitsbeweis und Laufzeitanalyse nicht vergessen!

Lösungsskizze

Wir konstruieren wie beim Algorithmus von Needleman und Wunsch eine Distanzmatrix D. Dabei beinhaltet der Eintag $D[i, j]$ die erweiterte Alignment-Distanz von $s_1 \cdots s_i$ mit $t_1 \cdots t_j$. Für die Rekursionsgleichungen erhalten wir dann für $i \geq 1$ und $j \geq 1$:

$$D[i, j] = \min \left\{ \begin{array}{l} D[i-1, j] + 1, \\ D[i, j-1] + 1, \\ D[i-1, j-1] + w(s_i, t_i), \\ D[i-2, j-2] + w^+(s_{i-1}s_i, t_{i-1}t_i) \end{array} \right\},$$

dabei gilt wiederum $w(x, x) = 0$ und $w(x, y) = w(x, -) = w(-, y) = 1$ für $x \neq y \in \Sigma$ sowie $w^+(xy, yx) = 1$ für $x \neq y \in \Sigma$. Der Eintrag $D[i-2, j-2] + w^+(s_{i-1}s_i, t_{i-1}t_i)$ wird dabei als ∞ interpretiert, wenn $i \leq 1$ oder $j \leq 1$.

Es stellt sich nun noch die Frage, welche Werte jeweils in der 1. Zeile bzw. in der 1. Spalte der Matrix stehen. Es gilt für $i \geq 0$ und $j \geq 0$:

$$D[0, j] = j \qquad \text{und} \qquad D[i, 0] = i$$

Die Korrektheit folgt nun aus der Tatsache, dass ein Alignment bzgl. der erweiterten Alignment-Distanz in der letzten Spalte entweder ein Indel, ein Match, eine Substitution oder in den letzten beiden Spalten eine Transposition besitzt.

Jeder Eintrag der Tabelle D lässt sich in konstanter Zeit ermitteln, von daher wird für das Erstellen der Tabellen Zeit $O(nm)$ benötigt. Die erweiterte Alignment-Distanz steht dann in $D[m, n]$.