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

Sommersemester 2016
Semestralklausur
21. Juli 2016

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

b) $T(n) = 2 \cdot T(n/4) + \sqrt{n} \log(n)$,

c) $T(n) = 3 \cdot T(n/3) + n\sqrt{n}$.

Vorname: ______________ Name: ______________ Matrikelnummer: ____________

Aufgabe 2 (8 Punkte)

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

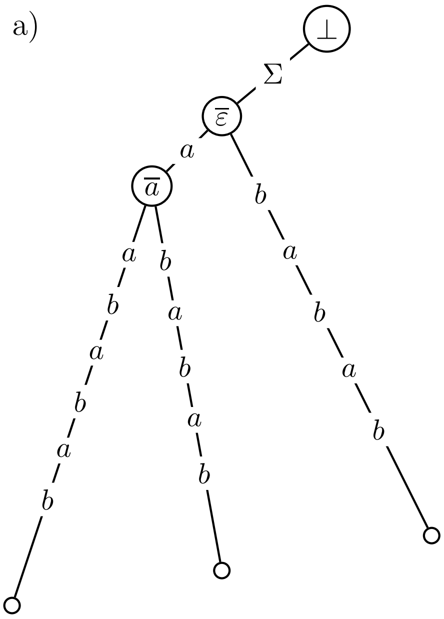
b) Gib die Kantenlabels so an, wie sie in Ukkonens Algorithmus verwendet werden.

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

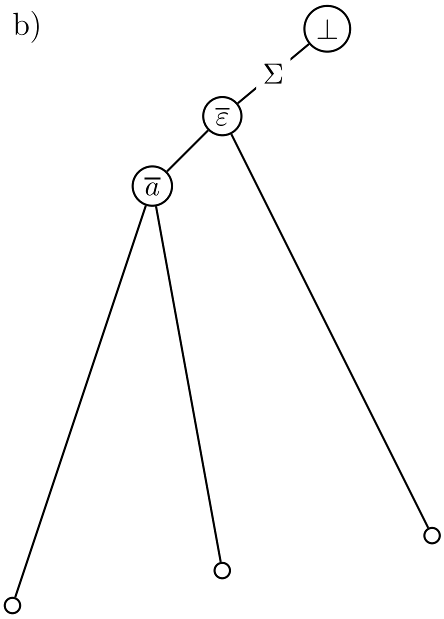
d) Führe Ukkonens Algorithmus für den Übergang von s' auf $s'' = s'{\cdot}a = aababababaa$ aus.

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.

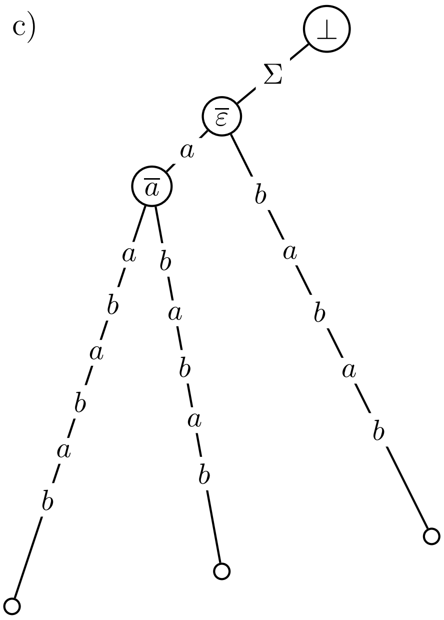
a)



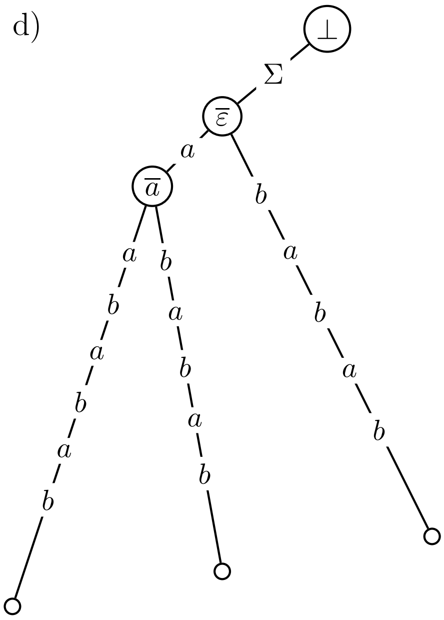
b)



c)



d)



Vorname: ______________ Name: ______________ Matrikelnummer: ____________

Aufgabe 3 (8 Punkte)

Löse die folgende Rekursionsgleichung **mit Hilfe der allgemeinen Lösung für lineare Rekursionsgleichungen**:

$$f_n = 4 \cdot f_{n-1} - 2 \quad \text{für } n \geq 1, \qquad \text{und} \qquad f_0 = 1.$$

Vorname: ______________ Name: ______________ Matrikelnummer: ____________

Aufgabe 4 (8 Punkte)

Gegeben seien zwei Wörter $s = s_1 \cdots s_m \in \Sigma^m$ und $t = t_1 \cdots t_n \in \Sigma^n$. Gib einen Algorithmus mit Laufzeit $O(n + m)$ an, der das längste Präfix von s findet, das auch ein Suffix von t ist.

Beispiel: Für $s = ababbaaaa$ und $t = bbbababb$ sind ε und $ababb$ jeweils sowohl ein Präfix von s als auch ein Suffix von t.

Hinweis: Korrektheitsbeweis und Laufzeitanalyse nicht vergessen!

Vorname: ____________ Name: ____________ Matrikelnummer: ____________

Aufgabe 5 (8 Punkte)

Ein *spezielles Alignment* für zwei Sequenzen $s \in \Sigma^n$ und $t \in \Sigma^m$ ist ein globales paarweises Sequenzen-Alignment $(\overline{s}, \overline{t}) \in \mathcal{A}(s,t)$ mit der Einschränkung, dass auf ein Indel (Insertion bzw. Deletion) keine Substitution folgen darf (also rechts davon stehen darf), jedoch ein Match oder ein Indel.

Beispiel: Für $s = AAAAC$ und $t = ATTC$ ist $\binom{AAA\text{−}AC}{\text{−}ATT\text{−}C}$ ein spezielles Alignment (allerdings nicht notwendigerweise ein optimales), $\binom{AA\boldsymbol{A}\text{−}AC}{A\text{−}\boldsymbol{T}T\text{−}C}$ oder $\binom{AAA\text{−}\boldsymbol{A}C}{A\text{−}\text{−}T\boldsymbol{T}C}$ jedoch nicht.

Finde einen möglichst effizienten Algorithmus, der für zwei gegebene Sequenzen $s \in \Sigma^n$ und $t \in \Sigma^m$ ein optimales spezielles Alignment bzgl. der Alignment-Distanz mit linearer Lückenstrafe findet. Hierbei trägt jede Insertion und jede Deletion 2 sowie jede Substitution 3 zur Alignment-Distanz bei, ein Match wie üblich 0.

Hinweis: Korrektheitsbeweis und Laufzeitanalyse nicht vergessen!