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

Wintersemester 2023/24
Semestralklausur
28. Februar 2024

Algorithmen auf Sequenzen

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							

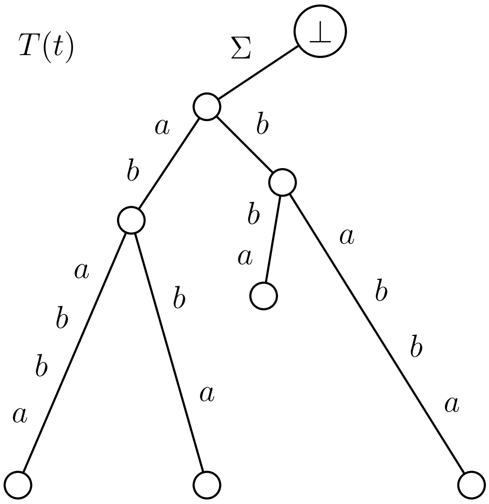
Vorname: ______________ Name: ______________ Matrikelnummer: ____________

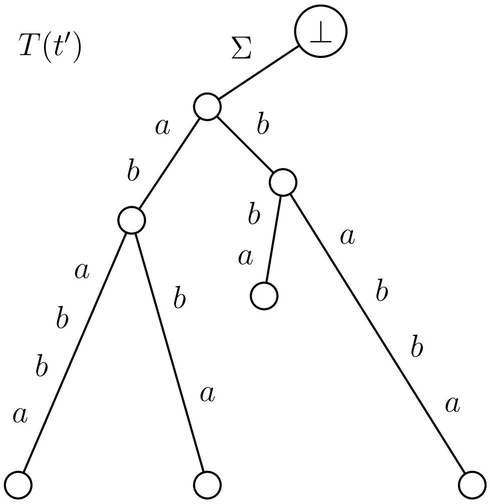
Aufgabe 1 (8 Punkte)

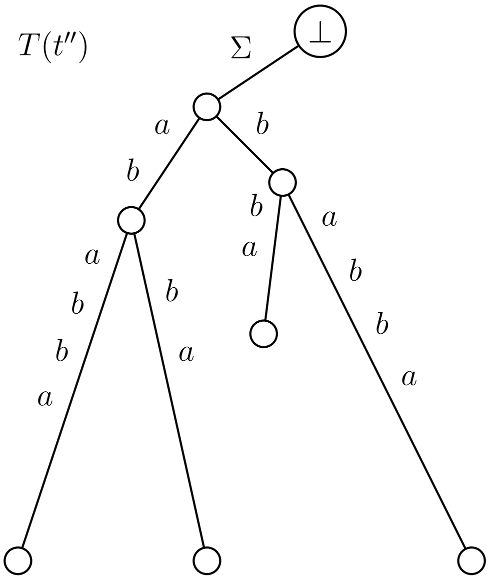
Zeichne im unteren Suffix-Baum $T(t)$ für $t = t_1 \cdots t_6 = ababba$ die Suffix-Links ein und erweitere diesen nach Ukkonens Algorithmus für $t' = ababba\underline{b}$, $t'' = ababba\underline{ba}$ und $t''' = ababba\underline{baa}$. Ergänze hierzu die unten angegebenen Suffix-Bäume.

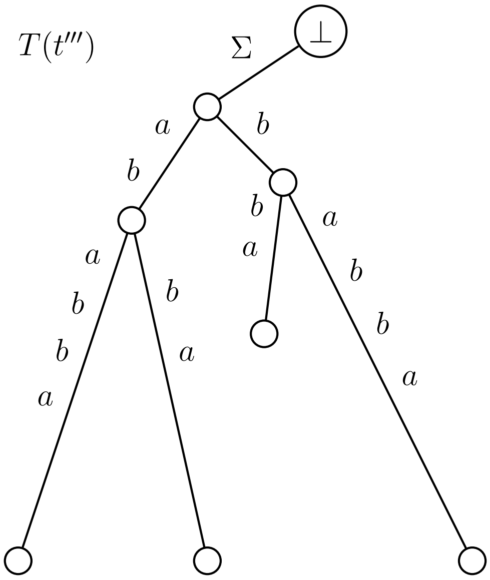
Es sind auch jeweils die neuen Suffix-Links und die Position des aktiven Suffixes sowohl vor als auch nach Ukkonens Erweiterungsschritt einzuzeichnen.

Gib für den Baum $T(t)$ (oben links) die Kantenmarkierungen an, die bei einer echten Implementierung hierfür verwendet werden.









Vorname: ____________ Name: ____________ Matrikelnummer: ____________

Aufgabe 2 (8 Punkte)

Betrachte das Wort $t\$ = t_1 \cdots t_{10}\$ = \text{ANANASANNA\$}$.

a) Konstruiere die Burrows-Wheeler-Transformierte $\hat{t}$ zu $t\$$.

b) Gib die zugehörige LF-Funktion für $\hat{t}$ an.

c) Bestimme die Werte $C(\cdot)$ und $Occ(\cdot,\cdot)$.

d) Suche nach $s = s_1 \cdots s_3 = \text{NNA}$ im FM-Index für t mit Hilfe des in der Vorlesung angegebenen Algorithmus unter Verwendung von C und Occ.

Hinweise: Für Teil a) und b) fülle die unten angegebene Tabelle korrekt aus. In dieser Aufgabe gilt: $\$ < A < N < S$.

i	$A[i]$	$t^{A[i]}$	$\hat{t}_i$	LF[i]
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Vorname: ____________ Name: ____________ Matrikelnummer: ____________

Aufgabe 3 (8 Punkte)

Erstelle für das Wort $t\$ = t_0 \cdots t_{10}\$ = ababbbababb\$$ ein Suffix-Array nach dem Algorithmus von Kärkkäinen und Sanders. Gib dabei alle Zwischenschritte an, wobei der rekursive Aufruf von Hand sortiert werden darf.

Hinweis: Gib beim Mischen von A_0 mit A_{12} für jede der Ergebnispositionen in A an, ob 1 oder 2 Zeichenvergleiche erforderlich waren oder ob auf die Ordnung von A_{12} zurückgegriffen wurde.

Vorname: ______________ Name: ______________ Matrikelnummer: ______________

Aufgabe 4 (8 Punkte)

Betrachte die unorientierte Permutation $\pi = (3, 4, 6, 5, 7, 8, 1, 2)$.

a) Zeichne den Breakpoint-Graphen $G(\pi)$ für π.

b) Gib die schärfere untere Schranke für die benötigte Anzahl von Reversionen zum Sortieren mit Reversionen aus der Vorlesung an und wende diese auf π bzw. den Breakpoint-Graphen $G(\pi)$ aus a) an.

c) Wende den Algorithmus aus der Vorlesung zur 2-Approximation für die minimale Reversal-Distanz auf π an. Gib dabei alle Zwischenschritte an und erläutere, warum eine bestimmte Reversion angewendet wird (bzw. nicht angewendet wird).

Vorname: ________________ Name: ________________ Matrikelnummer: ______________

Aufgabe 5 (8 Punkte)

Entwirf einen Linearzeit-Algorithmus für das folgende Problem.

MAXIMAL SCORING EVEN SUBSEQUENCE (MSES)

Eingabe: Eine Folge $(a_1, \ldots, a_n) \in \mathbb{R}^n$.

Ausgabe: Eine (zusammenhängende) Teilfolge $(a_\ell, \ldots, a_r)$ mit $\ell \leq r \in [1 : n]$, für die $r - \ell + 1$ gerade ist und die $\sigma(\ell, r)$ maximiert, wobei $\sigma(i, j) = \sum_{k=i}^{j} a_k$.

Hinweise: Korrektheitsbeweis und Laufzeitanalyse nicht vergessen.