

Universität Leipzig Institut für Informatik Automatische Sprachverarbeitung	Algorithmen und Datenstrukturen II SS 2018 – Serie 1		
Prof. Dr. Gerhard Heyer	Ausgabe am 18.04.2018	Abgabe am 25.04.2018	Seite 1/2

Algorithmen und Datenstrukturen II

SS 2018 – Serie 1

1 (10 Punkte) Huffman-Kodierung

Gegeben sei die Zeichenkette

$S_1 = \text{NESSELN_ESSEN_ESEL_NICHT_ESEL_ESSEN_NESSELN}$

- a) (1 Punkte) Zählen Sie die Häufigkeiten der Zeichen und geben Sie diese in einer Tabelle an. Versichern Sie sich, dass Ihre Zählung stimmt, damit Sie die beiden folgenden Teilaufgaben auch richtig lösen können.
- b) (2 Punkte) Konstruieren Sie anhand der in a) ermittelten Häufigkeiten den entsprechenden Huffman-Kodierungs-Baum. Gibt es mehr als zwei in Frage kommende Bäume, so werden diejenigen ausgewählt, die die am frühesten im Alphabet erscheinenden Zeichen enthalten. Die Zuordnung als linker und rechter Teilbaum bestimmt sich ebenfalls immer alphabetisch: der Baum mit dem früher vorkommenden Zeichen wird linker Teilbaum. Das Leerzeichen $_$ kommt alphabetisch vor A.
Hinweis: Es müssen keine Heap-Bäume oder PriorityQueues für die Sortierung gezeichnet werden.
- c) (2 Punkte) Geben Sie die Code-Werte jedes Zeichens an. Jedes Kantenlabel ist links mit 0 und jedes nach rechts mit 1 zu belegen. Kodieren Sie anschließend die ursprüngliche Zeichenkette.
- d) (5 Punkte) Erstellen Sie den Huffman-Kodierungs-Baum für die Zeichenkette $S_2 = _ \text{CEHILNST}$ und kodieren Sie die in a) gegebene Zeichenkette damit. Folgen Sie dem Vorgehen aus a), b) und c) und geben Sie die Zwischenschritte analog an. Welchen Unterschied stellen Sie nach der Kodierung fest?

2 (5 Punkte) LZW

Gegeben seien der Code $C_1 = 1\ 2\ 5\ 5\ 2\ 0\ 8\ 10\ 2\ 7$ und die initiale Codetabelle: 0: $_$ 1:B 2:I 3:E 4:T 5:L 6:R 7:G

Dekodieren Sie C_1 mit dem in der Vorlesung angegebenen LZW-Dekodierungsalgorithmus. Füllen Sie dabei folgende Tabelle:

gelesener Code	Präfix	Wort	Neues Wort	Erweiterung Code Tabelle	Ausgabe
1					
2					
...					

Universität Leipzig Institut für Informatik Automatische Sprachverarbeitung	Algorithmen und Datenstrukturen II SS 2018 – Serie 1		
Prof. Dr. Gerhard Heyer	Ausgabe am 18.04.2018	Abgabe am 25.04.2018	Seite 2/2

3 (5 Punkte) LZ77

Gegeben sei die Zeichenkette $S_3 = \text{WILLIBILLIBILLIGBIER}$

- a) (3 Punkte) Wenden Sie den Algorithmus LZ77 (“Sliding Window”-Lempel-Ziv ohne Optimierungen) auf S_3 an. Der Lookahead-Buffer sei 10 Zeichen lang und die Dictionarylänge sei 6. Protokollieren Sie in einer Tabelle pro Zeile: die kodierte Teilfolge, das erzeugte Codetriplett, den Inhalt von Dictionary-Window und Lookahead-Buffer. Die Tabelle enthalte eine Zeile pro erzeugtem Codetriplett.
- b) (2 Punkte) Gegeben sei folgender LZ77 Code: $(0,0,W), (0,0,I), (0,0,L), (1,1,I), (0,0,B), (2,1,L), (1,1,I), (0,0,B), (2,1,L), (1,1,I), (0,0,G), (0,0,B), (3,1,E), (0,0,R)$
Bestimmen Sie die Länge des für die Kodierung verwendeten Dictionary und begründen Sie ihre Angabe.

4 (5 Punkte) Burrows-Wheeler Transformation

- a) (3 Punkte) Gegeben sei die Zeichenkette $S_4 = \text{WILLIBILLI}$. Wenden Sie die Burrows-Wheeler-Transformation wie in der Vorlesung besprochen auf S_4 an. Alle Zähler beginnen bei 0. Geben Sie die alphabetisch sortierte Matrix der Rotationen von S_4 und die Ausgabe der Transformation (letzte Spalte und Zeilennummer des Originalblocks) an.
- b) (2 Punkte) Führen Sie die Rücktransformation durch. Hierbei ist es ausreichend, den Gesamtpfad zu zeichnen und Start- und Endpunkte zu markieren.