

Übungsblatt 6 – Lösungen

Aufgabe 1 - Huffman-Codierung

Gegeben sei der folgende Text:

gggbdgffgggcccfeeggcccbcccggeeffddgggaaaaddbddddeeffeefeeffeebeeeffgffbfiffgffgffgggffgggffgggbggg

Bestimme den Huffman-Code für den Text und ergänzen Sie die folgende Tabelle:

Zeichen	Häufigkeit	Huffman-Code
a	5	0100
b	6	0101
c	9	000
d	10	001
e	15	011
f	28	10
g	32	11

In einer 8-bit-ASCII-Codierung würde der Text 840 bits Speicherplatz benötigen.

Wie viele bits benötigt der Text in der oben bestimmten Huffman-Codierung?

$$32*2 + 28*2 + 15*3 + 10*3 + 9*3 + 6*4 + 5*4 = 266 \text{ (und die Tabelle mit der Codierung)}$$

Aufgabe 2 - Huffman-Decodierung

Gegeben sei die folgende Codierung:

1001001101011111100100011110111010101110000101

Bestimme den zugehörigen Text:

dap_macht_spass

Die Codierung der einzelnen Zeichen ist :

token: _ -> code: 101	token: a -> code: 00	token: c -> code: 1000
token: d -> code: 1001	token: h -> code: 11110	token: m -> code: 11111
token: p -> code: 110	token: s -> code: 01	token: t -> code: 1110

Aufgabe 3 - Huffman-Dekodierung

Erweitere die Klasse HuffmanTree um eine Methode `public String decode(String encoded)`, die den Huffman-kodierten Text `encoded` wieder in seine Originaldarstellung bringt. Gehe davon aus, dass die Methode auf genau dem Huffman-Baum aufgerufen wird, der auch zur Kodierung des Textes verwendet wurde. Verwende dabei eine Referenz HuffmanTree `current`, die zu Beginn auf die Wurzel des Baums verweist. Während `encoded` zeichenweise gelesen wird, wird die Referenz je nach Zeichen 0 oder 1 auf den linken oder den rechten Kindknoten gesetzt. Wird ein Blatt erreicht, wird der zugehörige Buchstabe dem Ergebnis hinzugefügt und die Referenz wieder auf die Wurzel des Baums gesetzt.

(Lösung als Java-Datei)