LU04c - Huffman

Siehe auch http://de.wikipedia.org/wiki/Datenkompression

Einleitung

Die Huffman-Codierung funktioniert ähnlich wie das Morsealphabet. Es wird aber kein fest vorgegebener Baum mit den Zeichen und Codes verwendet. Stattdessen wird die Codierung während der Verarbeitung erstellt. Dadurch wird für jede Art von Daten das häufigste Zeichen den kürzesten Code erhalten.

Vorgehen

Um das Vorgehen verständlich zu machen, verwenden wir ein Beispiel. Wir wollen den folgenden Text komprimieren: „bitte_nehmen_sie_ihren_abfall_mit“.

1. Notiere alle Zeichen und deren Häufigkeit

Jedes Zeichen wird genau 1x notiert. Als Knoten (Kreis) wird die Häufigkeit des Zeichens notiert.

2. Verbinde jeweils zwei Knoten

Wir suchen zwei Knoten mit den kleinsten Zahlen, die noch keine Verbindung nach oben haben. Diese Knoten verbinden wir zu einem neuen Knoten und addieren die Häufigkeit. Immer wenn sie eine Verbindung nach oben Ziehen, steht der Ast rechts für den binären Code „1“ und der linke Ast für den binären Code „0“.

3. Wiederhole

Der Schritt 2 wird wiederholt, bis nur noch ein Knoten übrig sind.

Fertiger Baum

Durch dieses Vorgehen wird gleichzeitig sichergestellt, dass die Codes eindeutig sind. Kein Code ist gleichzeitig der Anfang eines anderen Codes. Dadurch ist die Verwendung eines Trennzeichens

überflüssig.

Im Moodle-Kurs finden Sie das Lernprogramm: HuffmanShannonFano.jar. Mit diesem Programm können Sie das Erstellen von Huffman-Bäumen üben.

Zum Starten geben Sie in der Konsole ein:

```
java -jar HuffmanShannonFano.jar
```

Codierter Text

Codetabelle

Aus dem Baum lässt sich die Codetabelle mit den Zeichen und ihren Codes ableiten.

Zeichen	Häufigkeit	Code
_	5	110
e	5	111
i	4	100
n	3	000
t	3	001
a	2	0100
b	2	0101
h	2	0110
m	2	0111
l	2	1010
r	1	101100
f	1	101101
s	1	10111

Je häufiger das Zeichen im Text vorkommt, desto kürzer ist der Code.

Codierter Text

```
0101 100 001 001 111 110 000 111 0110 0111 111 000 110 10111 100 111 110 100
0110 101100 111 000 110 0100 0101 101101 0100 1010 1010 110 0111 100 001
```

Zur Veranschaulichung wurde hier nach jedem Zeichen ein Leerschlag eingefügt. In einer Huffman-codierten Datei würden alle Bits ohne Unterbruch hintereinander stehen:

```
0101100001001111110000111011001111110001101011110011111010001101011001110001
1001000101101101010010101010110011110000 1
```

Kompressionsfaktor

Vergleichen wir den Speicherplatz des ursprünglichen Textes mit dem komprimierten Code:

- `bitte_nehmen_sie_ihren_abfall_mit`: 33 ASCII-Zeichen à 8 Bit = 264 Bit
- Huffman-Code: 117 Bit

$$\text{Kompressionsfaktor} = 100 - \frac{117 * 100}{264} = 55.7\%$$

In diesem vereinfachten Beispiel haben wir den Speicherplatz für die Codetabelle ausser acht gelassen.

Decodieren

Zum Decodieren wird jeder Code durch das entsprechende Zeichen ersetzt. Dies wird erleichtert, wenn die Codetabelle nach dem Code aufsteigend sortiert ist.

Code	Zeichen
000	n
001	t
0100	a
0101	b
0110	h
0111	m
100	i
1010	l
101100	r
101101	f
10111	s
110	_
111	e

Aus dem codierten Text werden so viele Bits genommen, bis diese Bitkombination zu einem Code in der Tabelle passt:

```
010110000100111111000011101100111111000110101111001111101000110101100111000110010001011011010100101010101100111100001
```

1. 0 ⇒ Passt zu keinem Code
2. 01 ⇒ Passt zu keinem Code
3. 010 ⇒ Passt zu keinem Code
4. 0101 ⇒ Buchstabe „b“

Nun werden diese Bits aus dem codierten Text entfernt und das Verfahren wird mit den nächsten Bits wiederholt:

```
1000010011111100001110110011111100011010111100111110100011010110011100011001
0001011011010100101010101100111100001
```

1. 1 ⇒ Passt zu keinem Code
2. 10 ⇒ Passt zu keinem Code
3. 100 ⇒ Buchstabe „i“

Einsatz der Huffman-Codierung

Aktuelle Kompressionsprogramme nutzen mehrere Kompressionsverfahren nacheinander. Dabei wird oft als letzte Stufe die Huffman-Codierung angewandt.

Die Huffman-Codierung wird auch bei Bildformaten eingesetzt. Zum Beispiel verwendet das PNG-Format unter anderem eine Huffman-Codierung.

---

m114-A0G

Marcel Suter

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m114/learningunits/lu04/huffman?rev=1769631167

Last update: **2026/01/28 21:12**