

# CS 303: Praktische Informatik II

## Übungsblatt 5

Prof. Dr. Rainer Gemulla

Universität Mannheim, FSS 2023

### 1 Huffman-Code

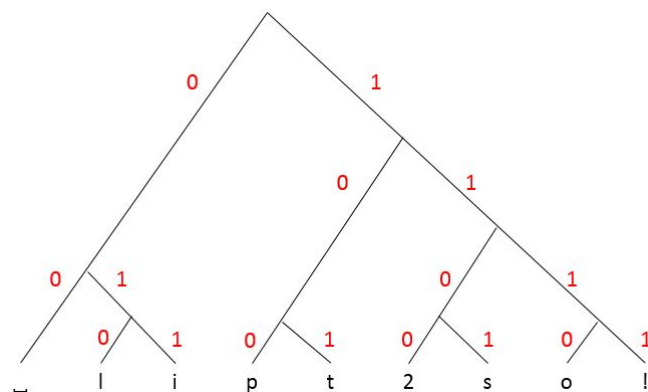
Gegeben sei die folgende Zeichenkette:

`password_sharp_sword`

- Wieviele Bits benötigt man mindestens pro Zeichen bei einer binären Kodierung mit fester Bitlänge? Kodieren Sie die Zeichenkette binär mit fester Bitlänge.
- Kodieren Sie die Zeichenkette binär mit dem Huffman-Algorithmus. Geben Sie anschließend die Kompressionsrate (Ersparnis) im Vergleich zu a) an.
- Angenommen, Sie erhalten die Kodierungen aus Aufgabenteil a) und b) mit je einem einzelnen Bit-Fehler an einer beliebigen Stelle. Wie wirkt sich dies jeweils auf die Dekodierung aus? Welchen Vorteil hat die Kodierung mit fester Bitlänge? Welchen Vorteil hat die Kodierung mit variabler Bitlänge?
- Gegeben sei folgende Kodierung

1000111100000111101101001011110010010001111

und der folgende Huffman-Baum. Welcher Satz ergibt sich daraus?



## 2 Kodierung

Betrachten Sie eine Nachrichtenquelle mit vier verschiedenen Zeichen. Zeichen A und B treten jeweils mit einer Wahrscheinlichkeit von 40% auf, Zeichen C oder D entsprechend mit einer Wahrscheinlichkeit von 20%. Die vier Zeichen werden wie folgt kodiert:

Zeichen A  $\Rightarrow$  10, Zeichen B  $\Rightarrow$  11, Zeichen C  $\Rightarrow$  110, Zeichen D  $\Rightarrow$  101.

- a) Wie gut ist der Code hinsichtlich des Platzbedarfs?
- b) Ist der Code ein Präfix-Code?
- c) Ist der Code dekodierbar?
- d) Finden Sie ein Beispiel für einen Code, welcher kein Präfix-Code, aber dennoch dekodierbar ist?

## 3 Entropie

- a) Wie ist Entropie definiert und welche Bedeutung hat sie für die Kodierung mit variabler Bitlänge?
- b) Gegeben sei die Zeichenkette `password_sharp_sword` aus Aufgabe 1 mit den relativen Häufigkeiten:

p: 10%, a: 10%, s: 20%, w: 10%, o: 10%, r: 15%, d: 10%, \_: 10%, h: 5%

Berechnen Sie die Entropie der entsprechenden Zufallsvariable.

- c) Vergleichen Sie das Ergebnis aus Aufgabenteil b) mit der erwarteten Codewortlänge für der Huffman-Kodierung aus Aufgabe 1. Diskutieren Sie!
- d) Wiederholen Sie Aufgabenteil b) und c) mit folgenden relativen Häufigkeiten:

p: 6,25%, a: 6,25%, s: 50%, w: 6,25%, o: 6,25%, r: 12,5%, d: 6,25%,  
\_: 6,25%, h: 0%

Diskutieren Sie!

- e) Betrachten Sie eine diskrete Zufallsvariable  $X$  über  $m$  Symbolen  $A = \{a_1, \dots, a_m\}$ , für die alle relativen Häufigkeiten Zweierpotenzen sind. Das heißt:

$$p_i = P(X = a_i) = 2^{-l_i} \quad \text{für } l_i \in \mathbb{N}_0.$$

Es gilt  $\sum_{i=1}^m p_i = 1$ . Ein Beispiel ist  $A_{ex} = \{a_1, a_2, a_3\}$  mit  $p_1 = 1/2$ ,  $p_2 = 1/4$  und  $p_3 = 1/4$ ; ein anderes Beispiel ist die relative Häufigkeitsverteilung aus Aufgabenteil d).

Wir behaupten: Bei der Verwendung der Huffman-Kodierung wird das Symbol  $a_i$  durch ein Codewort der Länge  $l_i$  dargestellt. Welche erwartete Codewortlänge ergibt sich dadurch? Wie verhält sie sich zur Entropie der Zufallsvariable  $X$ ?

- f) \*Beweisen Sie die Behauptung aus Aufgabenteil e).

## 4 Verfügbarkeit

Sie betreiben eine Infrastruktur, die in Abbildung 1 schematisch dargestellt ist und sich aus drei Einheiten mit einzelnen Komponenten zusammensetzt. Anfangs besitzen Sie 2 Prozessoren, 1 Festplatte und 5 Netzteile. Die Ausfallwahrscheinlichkeiten der jeweiligen Komponenten sind in Tabelle 1 gegeben.

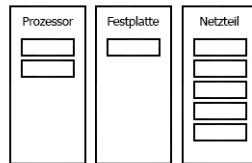


Abbildung 1: Darstellung der Einheiten mit Komponenten

Komponente	Ausfallwahrscheinlichkeit $x_i$
Prozessor	0.5%
Festplatte	1%
Netzteil	3%

Tabelle 1: Ausfallwahrscheinlichkeiten der Komponenten

- Das gesamte System fällt aus, wenn alle Komponenten einer Einheit ausfallen. Wie hoch ist die Wahrscheinlichkeit, dass dieser Fall eintritt?
- Sie möchten nun die Ausfallwahrscheinlichkeit des Systems auf 0.01% reduzieren. Wie viele Einheiten an Festplatten sollten Sie mindestens zukaufen, um die gewünschte Ausfallwahrscheinlichkeit zu erreichen?
- Redundanz ist eine Möglichkeit, um die Zuverlässigkeit eines Systems zu erhöhen. Welche Überlegungen sollten bei der Zusammenstellung eines Systems neben der Ausfallwahrscheinlichkeit der einzelnen Komponenten auch eine Rolle spielen?

## Projekt 4: Huffman-Kodierung

(10 Punkte)

- a) Erweitern Sie die Klasse `HuffmanCode.java`, die angeben soll, wie stark eine Datei mittels Huffman-Kodierung komprimiert werden kann. Wir betrachten dabei jedes Byte als ein Zeichen. Die Vorgabe stellt folgende Funktionalität zur Verfügung:
- Eine Datei wird von der Festplatte eingelesen (der Dateiname wird dem Programm als Argument übergeben) und die gelesenen Bytes in einem Array `char[] chars` gespeichert.
  - Die `main`-Methode liefert bereits den Rahmen für die erwartete Ausgabe.

Ihr Programm soll folgende Informationen über die Datei ausgeben:

- (i) aus wie vielen Zeichen die Datei besteht (0.5P)
- (ii) aus wie vielen verschiedenen Zeichen die Datei besteht (0.5P)
- (iii) wie viele Bits zur Kodierung der Datei durch einen optimalen Code fester Bitlänge benötigt werden (0.5P)
- (iv) wie viele Bits zur Kodierung der Datei durch den optimalen Huffman-Code benötigt werden (4P)
- (v) wie viel Speicherplatz prozentual im Vergleich zur ursprünglichen Kodierung (8 Bit pro Zeichen) gespart wird durch
  - i. die optimale feste Bitlänge, (0.5P)
  - ii. den Huffman-Code. (0.5P)
- (vi) die Shannon-Entropie der Datei (0.5P)
- (vii) welche 10 Zeichen am häufigsten auftreten. Geben Sie für jedes Zeichen die absolute und relative Häufigkeit, die Codelänge des Huffman-Codes, die hexadezimale Darstellung sowie für alle druckbaren Zeichen ( $\geq 33$ ) das Zeichen selbst aus. (1P)

**Hinweise.** Die Berechnung des tatsächlichen Huffman-Codes ist nicht gefordert. In diesem Projekt kann es hilfreich sein, zusätzliche Hilfsklassen *in derselben Java-Datei* zu implementieren.

**Optional.** Erweitern Sie ihr Programm, sodass es (1) die Huffman-Kodierung der Eingabedatei ausgibt und (2) eine so ausgegebene Datei wieder einlesen und dekodieren kann. Welche weiteren Informationen müssen dazu in der kodierten Datei abgelegt werden?

**Beispiel.** Für den Aufruf

```
java HuffmanCode blatt05-datei0.txt
```

soll ihr Programm folgende Ausgabe liefern:

```
Anzahl Zeichen                : 86
Anzahl verschiedener Zeichen   : 13
Kodierung mit fester Bitlänge  : 344 Bits (4 Bits pro Zeichen)
Kodierung mit Huffman-Code     : 303 Bits (3,52 Bits pro Zeichen)
Ersparnis (optimale feste Länge) : 50,00%
Ersparnis (Huffman-Code)       : 55,96%
Entropie                       : 3,50
```

Häufigste Zeichen:

```
0x61 a, Häufigkeit: 11 (12,8%), Codewortlänge: 3
0x0A  , Häufigkeit: 10 (11,6%), Codewortlänge: 3
0x0D  , Häufigkeit: 10 (11,6%), Codewortlänge: 3
0x62 b, Häufigkeit: 10 (11,6%), Codewortlänge: 3
0x63 c, Häufigkeit: 9 (10,5%), Codewortlänge: 3
0x64 d, Häufigkeit: 8 (9,3%), Codewortlänge: 4
0x65 e, Häufigkeit: 7 (8,1%), Codewortlänge: 4
0x66 f, Häufigkeit: 6 (7,0%), Codewortlänge: 4
0x67 g, Häufigkeit: 5 (5,8%), Codewortlänge: 4
0x68 h, Häufigkeit: 4 (4,7%), Codewortlänge: 4
```

- b) Wenden Sie ihr Programm auf die in ILIAS abgelegten Beispieldateien 1 bis 4 an. Alle Dateitypen sollen direkt eingelesen werden. Diskutieren Sie die Ergebnisse. Reichen Sie Ihre Programmausgaben und Diskussion bei der Projektabgabe in Form *einer PDF-Datei* innerhalb ihrer abzugebenden zip-Datei mit ein. (2P)