

Probeklausur

Grundlagen der Medieninformatik

Wintersemester 2018/19

Hinweise:

- Die Bearbeitungszeit beträgt 80 Minuten.
- 1 Punkt entspricht ungefähr einer Minute.
- Hilfsmittel sind keine erlaubt außer einem nicht programmierbarem Taschenrechner.
- Verwenden Sie einen dokumentenechten Stift.
- Bearbeiten Sie die Aufgaben alleine.
- Ungefähr eine Woche vor der ersten Klausur (06.02.2019) wird die Probeklausur im Plenum besprochen.
- Wenn Sie schon vorher Fragen haben oder Feedback zu Ihren Lösungen wollen, wenden Sie sich an Ihren Tutor oder kommen Sie in die Office Hour.

Viel
 \sum ☺
Erfolg

1 Huffman (10 Punkte)

1. Geben Sie für die Nachricht "schnelldurchlauf" den Zeichenvorrat mit der Wahrscheinlichkeitsverteilung an.

Zeichen x											
p(x)											

2. Zeichnen Sie den resultierenden Codebaum und beachten Sie dabei folgende Regeln:
 - Falls die Wahrscheinlichkeiten des linken und rechten Teilbaums gleich sind, wählen Sie als linken Teilbaum immer den Teilbaum, welcher das Blatt mit dem kleinsten ASCII-Wert besitzt.
 - Falls die Wahrscheinlichkeiten des linken und rechten Teilbaums ungleich sind, wählen Sie als linken Teilbaum immer den Teilbaum, welcher die kleinere Wahrscheinlichkeit besitzt.
3. Berechnen Sie die Durchschnittliche Wortlänge und Redundanz. Geben Sie dabei alle Rechenschritte an.

2 LZW-Kodierung (10 Punkte)

Kodieren Sie "fliegen_fliegen_hinter_fliegen" mittels der LZW-Kodierung.

Lesen(k)	Codetabelle schreiben (p & jk _i)	Ausgabe	Puffer füllen (p)
f			
l			
i			
e			
g			
e			
n			
_			
f			
l			
i			
e			
g			
e			
n			
_			
h			
i			
n			
t			
e			
r			
_			
f			
l			
i			
e			
g			
e			
n			
EOF			

3 Arithmetische Codierung (10 Punkte)

1. Vervollständigen Sie folgende Tabelle:

Zeichenindex i	1 = h	2 = e	3 = u	4 = a	5 = f	6 = n	7 = !
Häufigkeit p_i							
Linker Rand L_i							
Rechter Rand R_i							

Hierbei soll das Wort "heuhaufen!" aus dem nächsten Aufgabenteil als Basis für die Berechnung der Häufigkeit dienen.

2. Codieren Sie mit der vorangegangenen Tabelle das Wort "heuhaufen!" und geben Sie das Intervall sowie ein Ergebnis an. Stellen Sie die Zwischenergebnisse in einer Tabelle mit den Spalten Zeichen, L, R und B, ähnlich zu der in der Vorlesung dargestellten Tabelle, dar.

4 Arithmetische Decodierung (5 Punkte)

Decodieren Sie mit der selben Tabelle (siehe Aufgabe 3) die Darstellung 0.76165987 zu einem Wort der Länge 5 und geben Sie das Lösungswort an. Dokumentieren Sie Ihre Zwischenergebnisse und Ihren Lösungsweg.

5 HTML und CSS (15 Punkte)

1. Füllen Sie die Lücken in den HTML und CSS Dateien aus, so dass die Webseite aussieht wie auf dem Screenshot.

Hinweise:

- Es ist nicht erlaubt, bestehenden Code zu streichen oder zu ändern!
 - Es ist nicht erlaubt, neue Klassen (ids, tags) oder Blöcke in der CSS Datei hinzuzufügen.
 - Die CSS Datei heißt *style_klausur.css* und befindet sich im selben Ordner.
 - Es ist in der HTML und CSS Datei nur erlaubt, Code an den unterstrichenen Stellen einzufügen. Es müssen aber nicht alle Stellen genutzt werden.
 - Für volle Punktzahl müssen alle vorgegebenen Klassen der CSS Datei genutzt werden.
 - Das Bild befindet sich im selben Ordner, heißt *eule.jpg* und hat eine Größe von 400×200 .
 - Die generelle Schriftgröße beträgt *14pt*.
 - Der schwarze Rahmen um die Webseite gehört zu ihr und hat einen Radius (an den Ecken) von *10px*.
-
2. In der HTML-Datei haben sich zwei Fehler versteckt. Benennen Sie diese.

Screenshot



HTML Datei

```
<!DOCTYPE html>
<html>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="_____"/>
<body>
  <div class="box">
    <h1>
      <span class="_____">Probeklausuraufgabe HTML</span>
    </h1>
    In dieser Aufgabe lernen Sie wichtige Features von
    <span class="_____">HTML</span>
      und <span class="css">CSS</span>.
    </br>
    </br>
    <div class="blue-box">
      _____
    </br>
    </br>
    <div class="_____ ">
      
    </div>
  </br>
  <footer _____>
    Viel Glück beim Rest der Probeklausur.
  </footer>
</div>
</body>
</html>
```

CSS Datei

```
.html {  
    -----  
}  
  
body {  
    -----  
}  
  
.box {  
    -----  
    -----  
    -----  
    -----  
    -----  
}  
  
.css {  
    -----  
}  
  
.img-eule {  
    -----  
    -----  
}  
  
img {  
    -----  
    -----  
}
```

/* Es folgt noch mehr */

```
.blue-box {
    -----
    -----
    color: white;
    padding: 40px;
    font-size: 20px;
    -----
}

h1 span {
    /* Innenabstand ist 7px */
    -----
    -----
    -----
    -----
    -----
}

.bold {
    /* bold = fett */
    -----
}

.italic {
    /* italic = kursiv */
    -----
}

.underline {
    text-decoration: underline;
}

footer {
    /* Der footer hat eine Schriftgröße von 9px */
    -----
    -----
}
```

6 MPEG-Layer III Encoding (5 Punkte)

Beschriften (Nummerieren) Sie die Kästchen in folgender Abbildung mit den angegebenen Elementen eines MPEG-Layer-III-Encoders.

(Hinweis: Die Liste enthält einen Eintrag zu viel, dieser ist nicht im Encoder enthalten.)

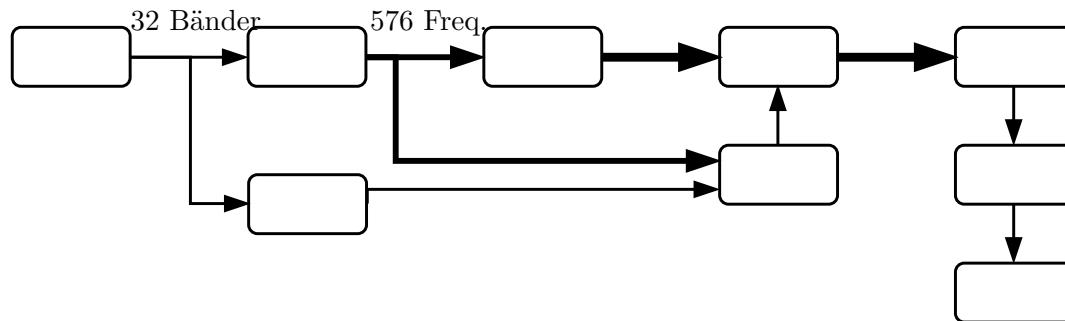


Figure 1:

Elemente eines MPEG-Layer-III-Encoders:

1. Komprimierte Daten
2. MDCT
3. Filterbank
4. DFT
5. FFT 1024
6. Bitstrom-Generator
7. PCM Audio
8. Huffman-Kodierung
9. Maskierung
10. Quantisierer

7 Fourier-Transformation (10 Punkte)

Führen Sie den reellen Teil der Fourier-Transformation für die Schwingung $f(t) = 7 \cdot \cos(2\pi t)$ durch. Das heißt, ignorieren Sie den Imaginärteil.

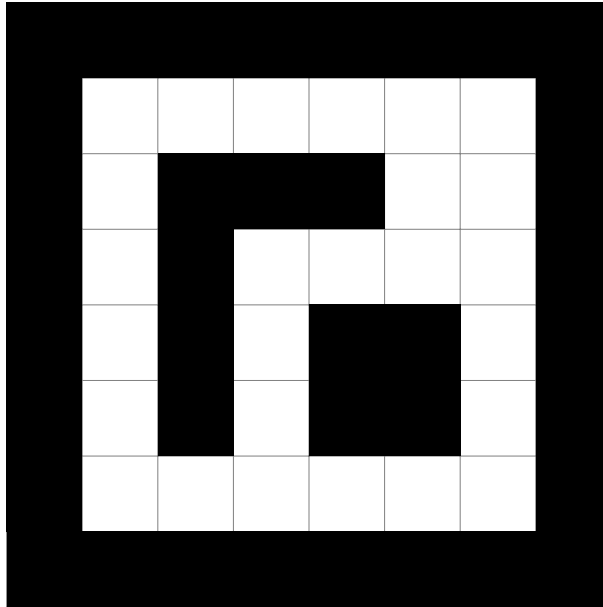
$$F(\omega) = \int f(t) \cos(2\pi\omega t) dt + i \int -f(t) \sin(2\pi\omega t) dt = \int f(t) e^{-2\pi i \omega t} dt$$

Dabei soll $\omega = 1$ und $t \in [0, 2\pi]$ sein. Hinweis: $\cos^2(x) = \frac{1}{2}(1 + \cos(2x))$

8 Bilder und Lauflängenkodierung (15 Punkte)

Gegeben sei folgender (verkleinerter) QR-Code:

Der Code wird als Bitmap (schwarz = 0, weiß = 1) der Größe 8×8 gespeichert.



1. Kodieren Sie das Bild mit Hilfe der Lauflängenkodierung auf zwei Arten.
 - a) Indem Sie zeilenweise (von links oben) durch das Bild gehen.

 - b) Indem Sie spaltenweise (von links oben) durch das Bild gehen.

 - c) Welche Vorgehensweise verbraucht weniger Speicher?
Begründen Sie Ihre Antwort.

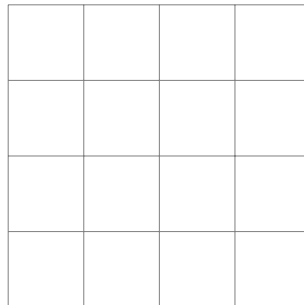
2. Betrachten Sie den folgenden Pseudo-Code und die darunter stehenden Hinweise:

```
scale8x8Image(img, scaledImage):  
  fuer alle x von 0 bis 3  
    fuer alle y von 0 bis 3  
      sum ← img(2x, 2y) + img(2x + 1, 2y)  
          + img(2x, 2y + 1) + img(2x + 1, 2y + 1)  
      wenn sum >= 2  
        scaledImage(x, y) ← 1  
  gebe scaledImage zurueck
```

Hinweise:

- In *img* steht das aktuelle Bild und *scaledImage* (4×4) soll beschrieben werden.
- *x* gibt die horizontale und *y* die vertikale Position an.
- Der Pixel (0,0) ist dabei der Pixel oben links.
- *img*(*x*, *y*) gibt in diesem Beispiel 0 für schwarz aus und 1 für weiß.
Z.B. *img*(5, 2) = 1 (weiß)

a) Zeichnen Sie den skalierten QR-Code in folgendes Raster:



b) Wie müsste der Pseudo-Code verändert werden, so dass ein Bild der Auflösung 99×99 nach 33×33 skaliert wird?