Probeklausur Besprechung

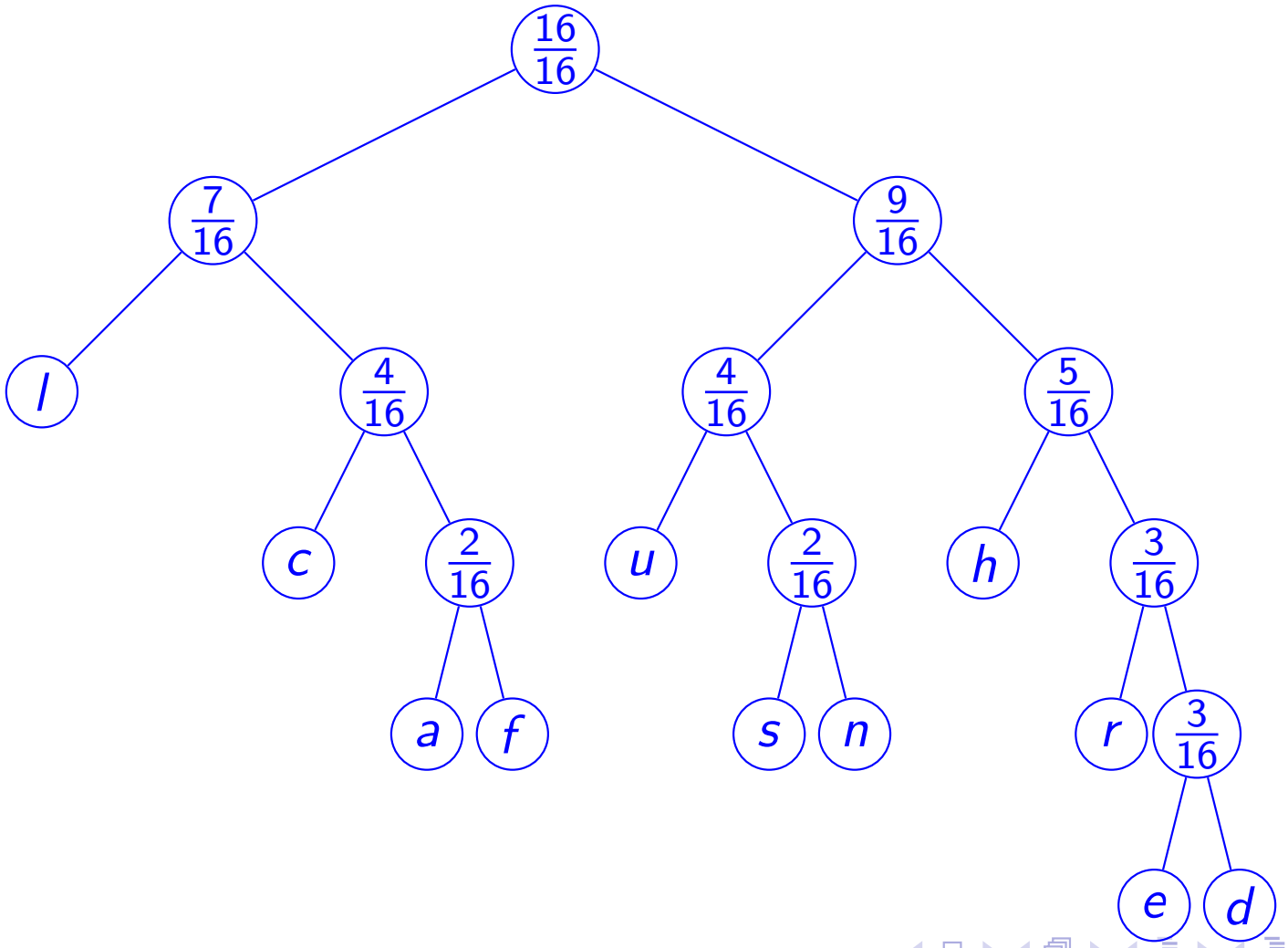
Grundlagen der Medieninformatik

WiSe 2018/19

Aufgabe 1: Huffman (10 Punkte)

1. Geben Sie für die Nachricht "schnelldurchlauf" den Zeichenvorrat mit der Wahrscheinlichkeitsverteilung an.
2. Zeichnen Sie den resultierenden Codebaum und beachten Sie dabei folgende Regeln:
3. Berechnen Sie die durchschnittliche Wortlänge und Redundanz. Geben Sie dabei alle Rechenschritte an.

Huffman Lösung



Wortlänge:

$$L_A = \sum_{a \in A} p_a \cdot |c(a)| = 3,375$$

Entropie:

$$H_A = \sum_{a \in A} p_a \cdot \log \frac{1}{p_a} = 3,3278195$$

Redundanz:

$$R_A = L_A - H_A$$

Aufgabe 2: LZW Kodierung (10 Punkte)

1. Kodieren Sie fliegen␣fliegen␣hinter␣fliegen mittels der LZW-Kodierung.

Lesen(k)	Codetabelle schreiben (p & ¡k¿)	Ausgabe	Puffer füllen (p)
f			
l			
i			
e			
g			
e			
n			
␣			
f			
l			
i			
e			
g			
e			
n			
␣			

Lesen(k)	Codetabelle schreiben (p & ¡k¿)	Ausgabe	Puffer füllen (p)
h			
i			
n			
t			
e			
r			
␣			
f			
l			
i			
e			
g			
e			
n			
EOF			

LZW Lösung

Lesen(k)	Codetabelle schreiben (p & <k>)	Ausgabe	Puffer füllen (p)
f			f
l	fl, 256	f	l
i	li, 257	l	i
e	ie, 258	i	e
g	eg, 259	e	g
e	ge, 260	g	e
n	en, 261	e	n
␣	n␣, 262	n	␣
f	␣f, 263	␣	f
l			fl
i	fli, 264	fl	i
e			ie
g	ieg, 265	ie	g
e			ie
n	gen, 266	ge␣	n

Lesen(k)	Codetabelle schreiben (p & <k>)	Ausgabe	Puffer füllen (p)
␣			n␣
h	n␣h, 267	n␣	h
i	hi, 268	h	i
n	in, 269	i	n
t	nt, 270	n	t
e	te, 271	t	e
r	er, 272	e	r
␣	r␣h, 273	r	␣
f			␣f
l	␣fl, 274	␣f	l
i			li
e	lie, 275	li	e
g			eg
e	ege, 276	eg	e
n			en
EOF		en	EOF

Aufgabe 3: Arithmetische Kodierung (10 Punkte)

1. Vervollständigen Sie folgende Tabelle für das Wort "heuhaufen":

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							

2. Kodieren 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.

Arithmetisch Kodierung Lösung

Initiale Tabelle:

Zeichenindex i	1 = h	2 = e	3 = u	4 = a	5 = f	6 = n	7 = !
Häufigkeit p_i	0.2	0.2	0.2	0.1	0.1	0.1	0.1
Linker Rand L_i	0.0	0.2	0.4	0.6	0.7	0.8	0.9
Rechter Rand R_i	0.2	0.4	0.6	0.7	0.8	0.9	1.0

Wiederhole :

$$a \leftarrow nextchar()$$

$$B \leftarrow R - L$$

$$L \leftarrow L + B \cdot L_a$$

$$R \leftarrow L + B \cdot R_a$$

Im Folgenden ist zu beachten, dass B für die jeweils nächste Zeile gilt.

Zeichen	L	R	B
h	0.0	0.2	0.2
e	0.04	0.08	0.04
u	0.056	0.064	0.008
h	0.056	0.0576	0.0016
a	0.05696	0.05712	0.00016
u	0.057024	0.057056	0.000032
f	0.0570464	0.0570496	0.0000032
e	0.05704704	0.05704768	0.00000064
n	0.057047552	0.057047616	0.000000064
!	0.0570476096	0.057047616	0.0000000064

Aufgabe 4: Arithmetische Dekodierung (5 Punkte)

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

Zeichenindex i	1 = h	2 = e	3 = u	4 = a	5 = f	6 = n	7 = !
Häufigkeit p_i	0.2	0.2	0.2	0.1	0.1	0.1	0.1
Linker Rand L_i	0.0	0.2	0.4	0.6	0.7	0.8	0.9
Rechter Rand R_i	0.2	0.4	0.6	0.7	0.8	0.9	1.0

Arithmetische Dekodierung Lösung

Man muss auf die Idee kommen, um das Intervall zwischen 0.76 und 0.762 auf n zu kommen indem man sich denkt, $16/2 = 8$ und 0.8 liegt im Intervall von n. Das Lösungswort lautet: fahne

Zeichen	L	R	B
f	0.7	0.8	0.1
a	0.76	0.77	0.01
h	0.76	0.762	0.002
n	0.7616	0.7618	0.0002
e	0.76164	0.76168	0.00004

Aufgabe 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.
2. In der HTML-Datei haben sich zwei Fehler (bezüglich XHTML-Standard) versteckt. Benennen Sie diese.

Probeklausuraufgabe HTML

In dieser Aufgabe lernen Sie wichtige Features von HTML und CSS.

Frohe Weihnachten!

Viel Glück beim Rest der Probeklausur.

- 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 $10px$.

HTML-Code

```
<!DOCTYPE html>
<html>
<meta charset=ÜTF-8¨>
<link rel=’’stylesheet’’ type=’’text/css’’  href=’’__________________’’></link>
<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=’’__________________’’>
                <img src=’’__________________’’></img>
            </div>
        </br>
    <footer __________________>
        Viel Glück beim Rest der Probeklausur.
    </footer>
    </div>
</body>
</html>
```

CSS-Code Teil 1

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

                              /* Es folgt noch mehr */
```

CSS-Code Teil 2

```
.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 */
    -----------------
    -----------------
}
```

HTML Lösung

```
<!DOCTYPE html>
<html>
<meta charset=ÜTF-8¨>
<link rel=’’stylesheet’’ type=’’text/css’’  href=’’style_klausur.css’’></link>
<body>
    <div class=’’box’’>
        <h1>
            <span class=’’’underline’’>Probeklausuraufgabe HTML</span>
        </h1>
        In dieser Aufgabe lernen Sie wichtige Features von
        <span class=’’’html’’>HTML</span>
            und <span class=’’css’’>CSS</span>.
        </br>
        </br>
        <div class=’’blue-box’’>
            Frohe Weihnachten
            </br>
            </br>
            <div class=’’img-eule" alt="eule">
                <img src=’’’eule.jpg’’></img>
            </div>
        </br>
        </div>
    <footer ___________________>
        Viel Glück beim Rest der Probeklausur.
    </footer>
    </div>
</body>
</html>
```

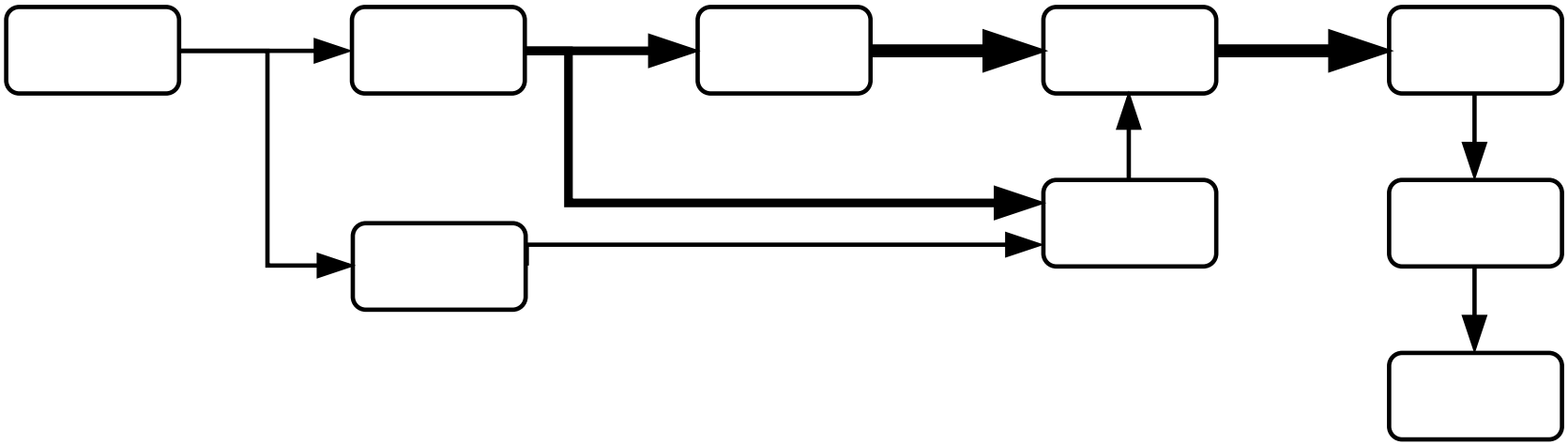
CSS Lösung 1

```
.html {
    color: red;
}

body {
    font-size: 14pt;
}

.box {
    border: solid;
    border-radius: 10px;
    border-color: black;
    padding 20px;
    -----------------
    -----------------

}

.css {
    color: green;
}

.img-eule {
    text-align: center;
    -----------------
}

img {
    width: 400px;
    height: 200px;
}

                                /* Es folgt noch mehr */
```

CSS Lösung Teil 2

```
.blue-box {
    background-color: blue;
    border-radius: 15px;
    color: white;
    padding: 40px;
    font-size: 20px;
    -----------------
}

h1 span {
    /* Innenabstand ist 7px */
    background-color: red;
    padding: 7px;
    color: white;
    font-weight: bold;
    -----------------
    -----------------
}

.bold {
    /* bold = fett */
    font-weight: bold;
}

.italic {
    /* italic = kursiv */
    font-style: italic;
}
.underline {
    text-decoration: underline;
}

footer {
    /* Der footer hat eine Schriftgröße von 9px */
    font-size: 9pt;
    text-align: right;
}
```

Aufgabe 6: MPEG-Layer III Encoding (5 Punkte)

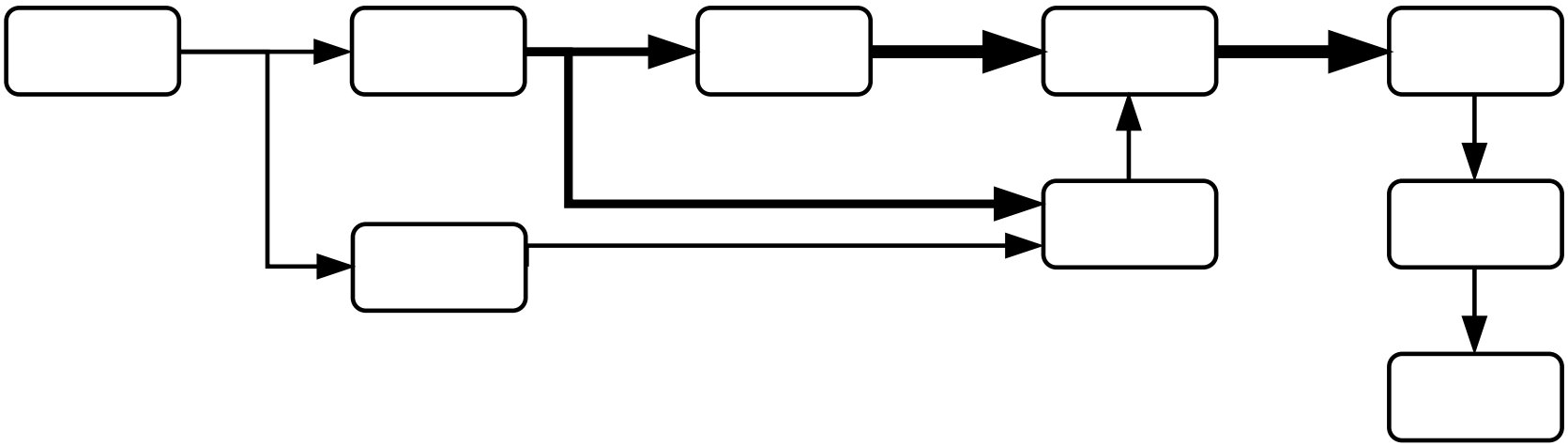


1. Beschriften (Nummerieren) Sie die Kästchen in folgender Abbildung mit den angegebenen Elementen eines MPEG-Layer-III-Encoders.
   (Hinweis: Die Liste enthält einen Eintrag zu viel, dieser ist nicht im Encoder enthalten.)

Elemente eines MPEG-Layer-III-Encoders:

- Komprimierte Daten
- MDCT
- Filterbank
- DFT
- FFT 1024
- Bitstrom-Generator
- PCM Audio
- Huffman-Kodierung
- Maskierung
- Quantisierer

MPEG-Layer III Encoding Lösung



Erste Zeile: PCM Audio, Filterbank, MDCT, Quantisierer, Huffman-Kodierung
Zweite Zeile: FFT 1024, Maskierung, Bitstrom-Generator
Dritte Zeile: Komprimierte Daten

Aufgabe 7: Fourier-Transformation (10 Punkte)

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

$$F(\omega) = \int f(t) \cos(2\pi\omega t) dt$$

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

Fourier-Transformation Lösung

$$F(1) = \int_0^{2\pi} 7 * \cos(2t\pi)\cos(2\pi t)dt$$

$$\Leftrightarrow F(1) = 7\int_0^{2\pi} \cos^2(2t\pi)dt$$

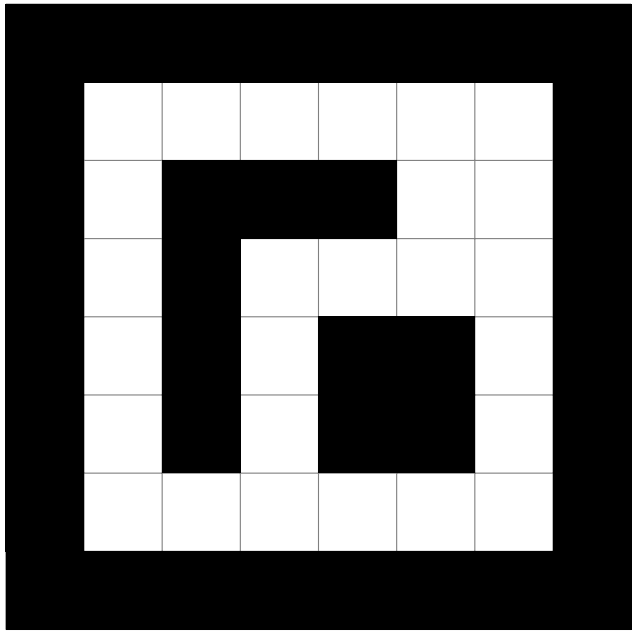
$$\Leftrightarrow F(1) = \frac{7}{2}\int_0^{2\pi} (1 + \cos(4\pi t))dt$$

$$\Leftrightarrow F(1) = \frac{7}{2}[x + \frac{1}{4\pi}sin(4\pi t)]_0^{2\pi}$$

$$\Leftrightarrow F(1) = 7(\pi + \frac{\sin(8\pi^2)}{8\pi})$$

Aufgabe 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.



Algorithm 1 Scale8x8Image(img, scaledImage)

```
1: for x=0 to 3 do
2:     for y=0 to 3 do
3:         sum ← img(2x, 2y) + img(2x + 1, 2y)
               + img(2x, 2y + 1) + img(2x + 1, 2y + 1)
4:         if sum ≥ 2 then
5:             scaledImage(x, y) ← 1
6:         end if
7:     end for
8: end for
9: return scaledImage
```

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)

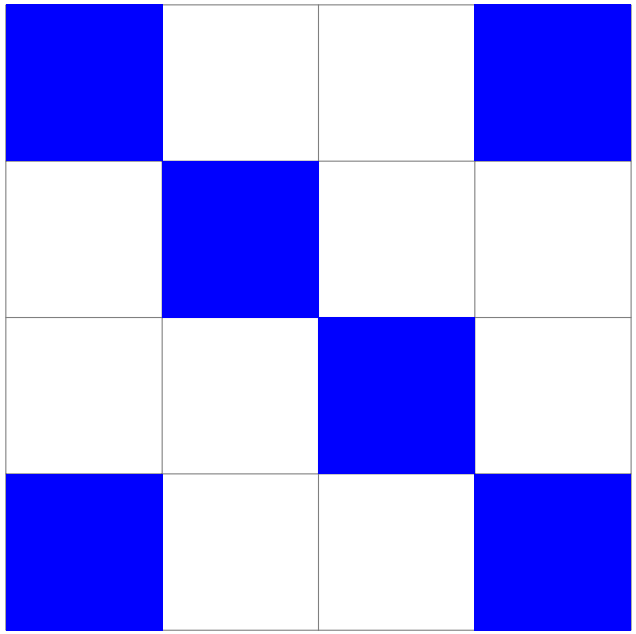
1. Kodieren Sie das Bild mit Hilfe der Lauflängenkodierung auf zwei Arten.
   - Indem Sie zeilenweise (von links oben) durch das Bild gehen.
   - Indem Sie spaltenweise (von links oben) durch das Bild gehen.
   - Welche Vorgehensweise verbraucht weniger Speicher? Begründen Sie Ihre Antwort.
2. Zeichnen Sie den skalierten QR-Code in folgendes Raster...
3. Wie müsste der Pseudo-Code verändert werden, so dass ein Bild der Auflösung 99×99 nach 33×33 skaliert wird?

Bilder und Lauflängenkodierung Lösung

Lauflängenkodierung

- #09 #16 #02 1 #03 #12 #02 1 0 #14 #02 1 0 1 #02 1 #02 1 0 1 #02 1 #02 #16 #09
- #09 #16 #02 1 #04 1 #02 1 0 #14 #02 1 0 1 #02 1 #02 #13 #02 1 #02 #16 #09
- zeilenweise 53 > spaltenweise 51 Zeichen

Skaliertes Bild:



Veränderter Pseudo-Code:

Algorithm 2 Scale99x99Image(img, scaledImage)

1: **for** x=0 to 32 **do**
2: **for** y=0 to 32 **do**
3: $sum \leftarrow img(3x, 3y) + img(3x+1, 3y) + img(3x+2, 3y)$
$+ img(3x, 3y+1) + img(3x+1, 3y+1)$
$+ img(3x+2, 3y+1) + img(3x, 3y+2)$
$+ img(3x+1, 3y+2) + img(3x+2, 3y+2)$
4: **if** $sum \geq 4.5$ **then**
5: $scaledImage(x, y) \leftarrow 1$
6: **end if**
7: **end for**
8: **end for**
9: **return** *scaledImage*