Karlsruhe Institute of Technology

Institut für Technik der Informationsverarbeitung

Dipl.-Ing. J. Heißwolf

heisswolf@kit.edu

Karlsruher Institut für Technologie (**KIT**)

Digitaltechnik

2. Übung

Offene Fragen

Aus der letzten Übung:
Wie viele **gültige Codewörter** (gCW) ergeben sich bei einer **Hamming-Distanz** (HD) von **2 Bit** => Formel?

Gesamtzahl der **CW** für Bitbreite **n**: $\# \, CW = 2^n$

Anzahl der **gültigen CW** für Bitbreite n: $\# \, gCW = \frac{2^n}{2} = 2^{n-1}$

Erklärung:

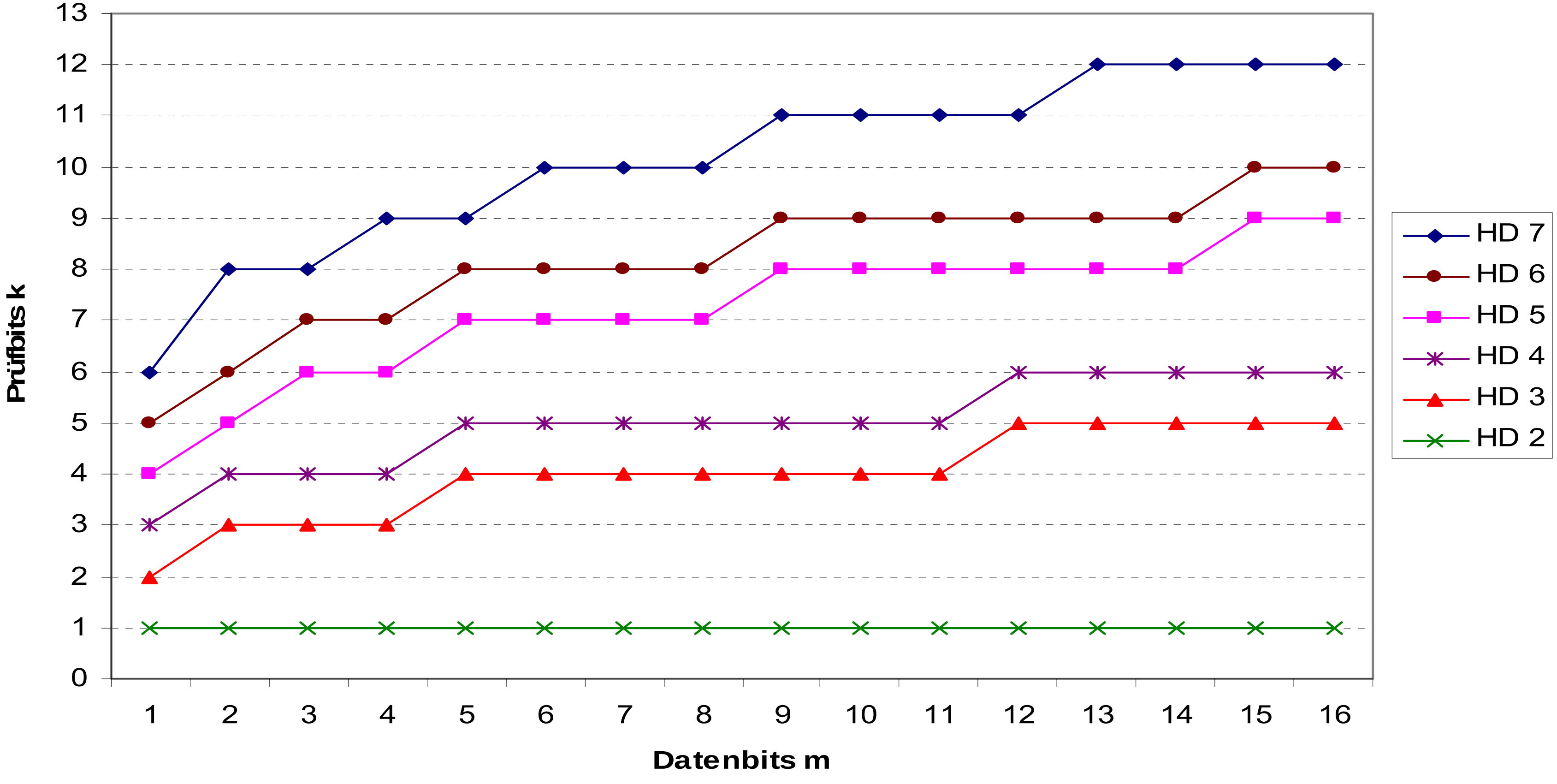
N-1 Datenbits	Prüfbit

Beispiel: Menge der CW für n=3 :
{{000, 011, 101, 110}, {001, 010, 100, 111}}
=> Anzahl der gCW: 4

Achtung: Für **größere Hamming-Distanzen** ergeben sich **komplexere Abhängigkeiten** zwischen der Anzahl der gCW und der Bitbreite n
Siehe Tafelanschrieb in VL: für HD = 3 und HD = 5

Offene Fragen

Aus der letzten Übung:
Wie viele **gültige Codewörter** (gCW) ergeben sich bei einer **Hamming-Distanz** (HD) von **2 Bit** => Formel?



Offene Fragen

Aus der letzten Vorlesung

JPEG-Verfahren: Kodierung (2) – Run-Length Kodierung

- ZigZag-geordnete AC-Koeffizienten werden so kodiert, dass nur AC-Koeffizienten ungleich Null in Binärvektorform (RRRRSSSS) dargestellt werden, wobei:
  - RRRR (4-Bit) -> Anzahl der Nullen vor dem aktuellen Koeffizienten angibt
  - SSSS (4-Bit) -> Wertebereiche der Koeffizienten gemäß unterer Tabelle repräsentiert

- der SSSS-Subvektor bestimmt höherwertige Bits der AC-Koeffizienten, die **niederwertigen Bits werden unkodiert an das RS-Symbol angehängt**

DC Coef Difference	Size	Typical Huffman codes for Size	Additional Bits (in binary)
0	0	00	-
-1,1	1	010	0,1
-3,-2,2,3	2	011	00,01,10,11
-7,...,-4,4,...,7	3	100	000,...,011,100,...111
-15,...-8,8,...,15	4	101	0000,...,0111,1000,...,1111
⋮	⋮	⋮	⋮
-1023,...-512,512,...,1023	10	1111 1110	00 0000 0000,...,11 1111 1111
-2047,...-1024,1024,...2047	11	1 1111 1110	000 0000 0000,...,111 1111 1111

RRRR	SSSS	Huffman-Code-Wort
0000	0000	1010 (EOB)
0000	0001	00
0000	0010	01
0000	0011	100
0000	0100	1011
0000	0101	11010
0000	0110	1111000
0000	0111	11111000
0000	1000	1111110110
0000	1001	1111111110000010
0000	1010	1111111110000011

1. Aufgabe

Einige Studenten wollen ihre Verabredungen über das Internet koordinieren.

Zu diesem Zweck legen sie den Zeitpunkt fest, nur über den Treffpunkt kommt keine Einigkeit zustande.

Um nun die Übertragungszeiten gering zu halten, beschließen sie, die beliebtesten Treffpunkte im Fanø–Code zu kodieren.

1. Aufgabe

1.1 Die Gaststätten sollen in “billig” und “teuer” eingeteilt werden. Wie groß ist der Informationsgehalt der Aussage “billig”, wenn “billig” Preise bis max. € 3,00 beinhaltet?

Gaststätte	Bevorzugt von Anzahl an Studenten	Kosten pro Getränk / €
Brasil	1	4,00
Dom	2	4,00
Krokodil	1	3,50
Miro	2	3,00
Titanic	6	2,50
Ubu	6	3,00
Wien	5	3,50
Zero	3	3,50

1. Aufgabe

1.1 Die Gaststätten sollen in “billig” und “teuer” eingeteilt werden. Wie groß ist der Informationsgehalt der Aussage “billig”, wenn “billig” Preise bis max. € 3,00 beinhaltet?

Gaststätte	Bevorzugt von Anzahl an Studenten	Kosten pro Getränk / €
Brasil	1	4,00
Dom	2	4,00
Krokodil	1	3,50
Miro	2	3,00
Titanic	6	2,50
Ubu	6	3,00
Wien	5	3,50
Zero	3	3,50

1. Aufgabe

1.1 Die Gaststätten sollen in “billig” und “teuer” eingeteilt werden. Wie groß ist der Informationsgehalt der Aussage “billig”, wenn “billig” Preise bis max. € 3,00 beinhaltet?

$$p = \frac{\sum \text{billige Kneipen}}{\sum \text{aller Kneipen}} = \frac{3}{8} = 0{,}375$$

$$H_e = \mathrm{ld}\,(1/p) = \mathrm{ld}\;(1/0{,}375) = \underline{\underline{1{,}42\ \text{bit}}}$$

KIT – Universität des Landes Baden-Württemberg und nationales Forschungszentrum in der Helmholtz-Gemeinschaft
Institut für Technik der Informationsverarbeitung

1. Aufgabe

Karlsruhe Institute of Technology

1.2 Berechnen Sie die Wahrscheinlichkeit für den Besuch jeder einzelnen Gaststätte. Gehen Sie davon aus, dass die Wahrscheinlichkeit von der Beliebtheit abhängig ist.

Gaststätte	Bevorzugt von	Kosten pro Getränk / €
Brasil	1	4,00
Dom	2	4,00
Krokodil	1	3,50
Miro	2	3,00
Titanic	6	2,50
Ubu	6	3,00
Wien	5	3,50
Zero	3	3,50
	$\Sigma = 26$	

Auftrittswahrscheinlichkeit:
p = Einzel El. / Gesamt

$p(\text{Brasil}) = 1 / 26 \approx 0.0385$

$p(\text{Dom}) = 2 / 26 \approx 0.077$

$p(\text{Krokodil}) = 1 / 26 \approx 0.0385$

$p(\text{Miro}) = 2 / 26 \approx 0.077$

$p(\text{Titanic}) = 6 / 26 \approx 0.231$

$p(\text{Ubu}) = 6 / 26 \approx 0.231$

$p(\text{Wien}) = 5 / 26 \approx 0.192$

$p(\text{Zero}) = 3 / 26 \approx 0.115$

$\Sigma = 1$

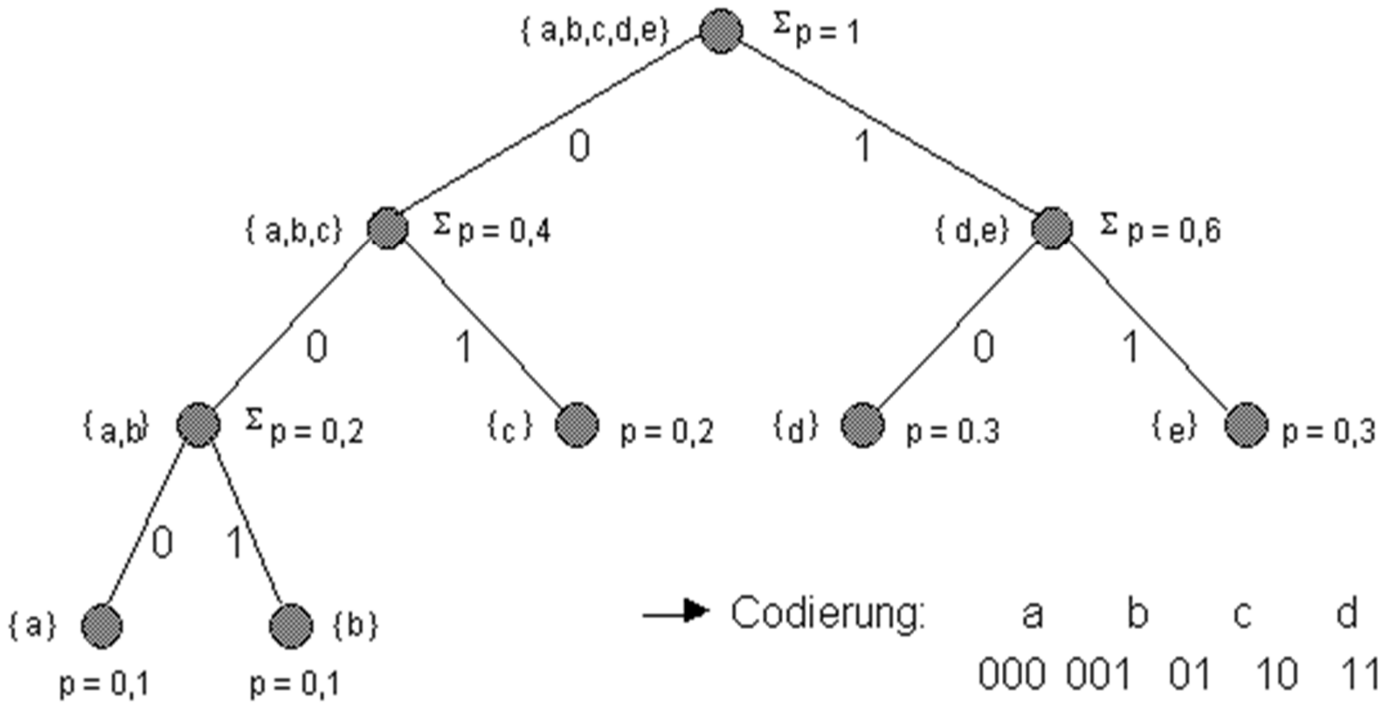
1. Aufgabe

1.3 Codieren Sie die Zeichen in Fan∅-Code.

FanØ-Codierung

Beispiel:

Zeichen:	a	b	c	d	e
Wahrscheinlichkeit p:	0,1	0,1	0,2	0,3	0,3



→ Codierung:

a	b	c	d	e
000	001	01	10	11

KIT – Universität des Landes Baden-Württemberg und nationales Forschungszentrum in der Helmholtz-Gemeinschaft
Institut für Technik der Informationsverarbeitung

1. Aufgabe

Gaststätte	Bevorzugt von	Kosten pro Getränk / €	p
Brasil	1	4,00	0.0385
Dom	2	4,00	0.077
Krokodil	1	3,50	0.0385
Miro	2	3,00	0.077
Titanic	6	2,50	0.231
Ubu	6	3,00	0.231
Wien	5	3,50	0.192
Zero	3	4,50	0.115

sortieren nach p!

Gast-stätte	Titanic	Ubu	Wien	Zero	Dom	Miro	Brasil	Krokodil
p	0,231	0,231	0,192	0,115	0,077	0,077	0,0385	0,0395

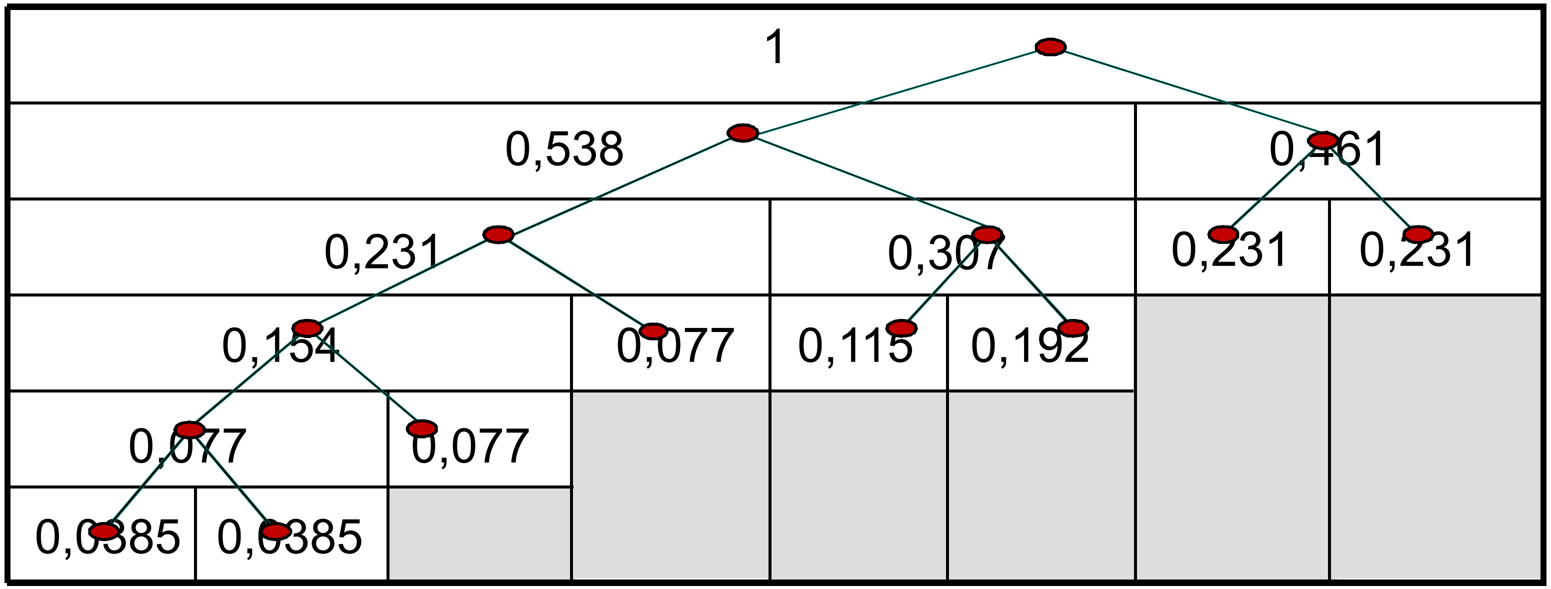
1. Aufgabe

Gast-stätte	Krokodil	Brasil	Miro	Dom	Zero	Wien	Ubu	Titanic
p	0,0385	0,0385	0,077	0,077	0,115	0,192	0,231	0,231

<table>
<tr><td colspan="8">1</td></tr>
<tr><td colspan="6">0,538</td><td colspan="2">0,461</td></tr>
<tr><td colspan="4">0,231</td><td colspan="2">0,307</td><td>0,231</td><td>0,231</td></tr>
<tr><td colspan="3">0,154</td><td>0,077</td><td>0,115</td><td>0,192</td><td></td><td></td></tr>
<tr><td colspan="2">0,077</td><td>0,077</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>0,0385</td><td>0,0385</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

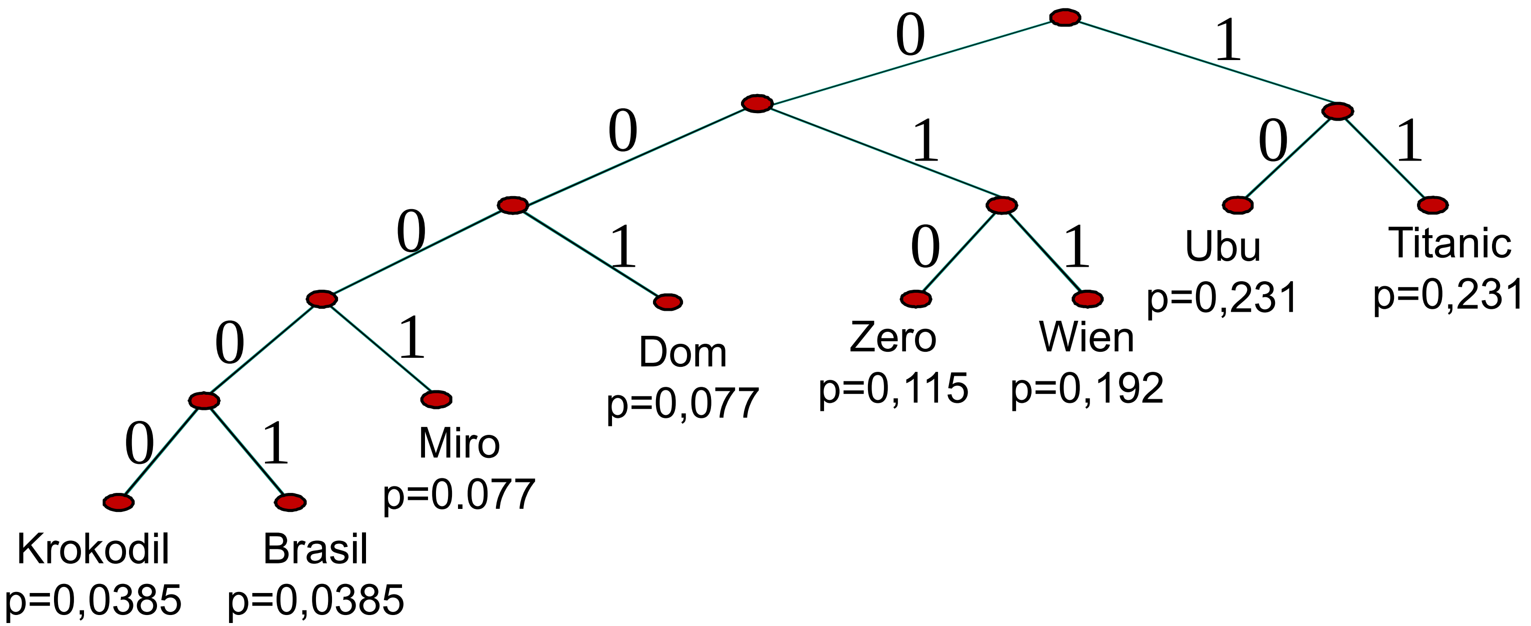
1. Aufgabe

Gast-stätte	Krokodil	Brasil	Miro	Dom	Zero	Wien	Ubu	Titanic
p	0,0385	0,0385	0,077	0,077	0,115	0,192	0,231	0,231



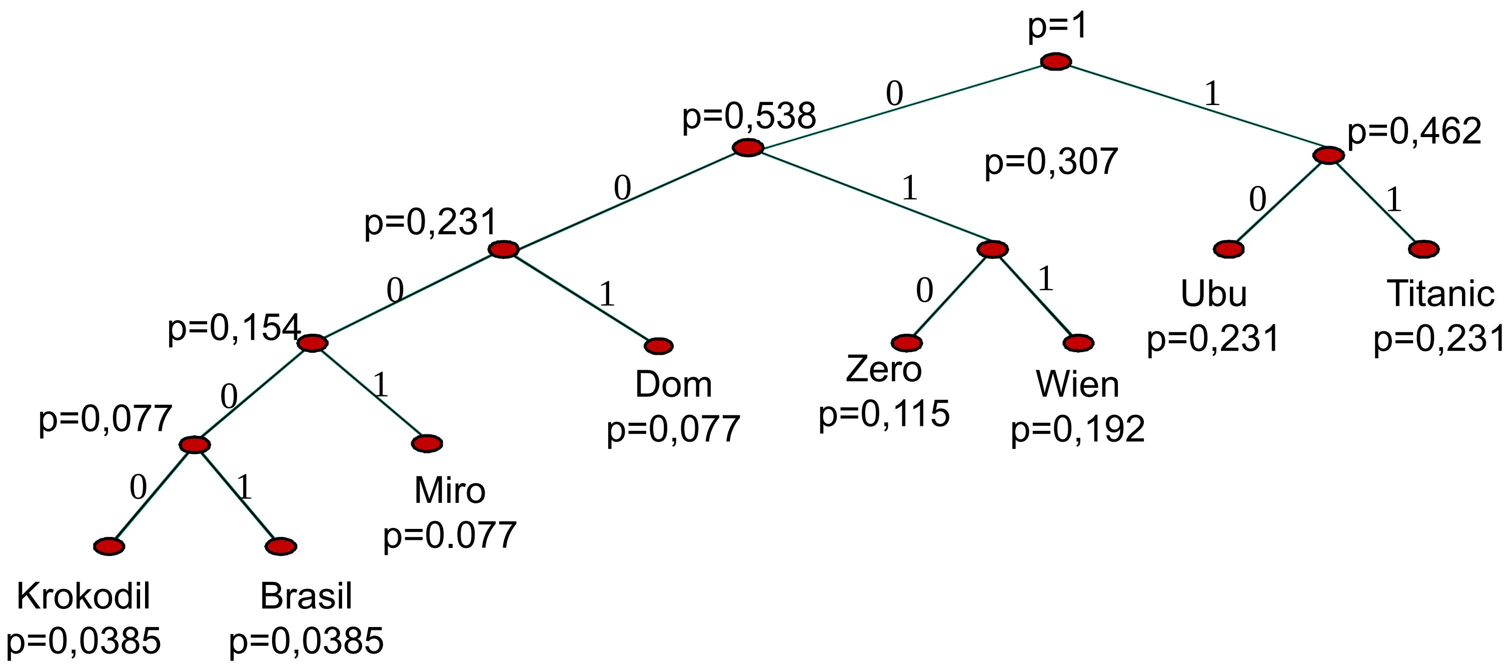
1. Aufgabe

Gast-stätte	Krokodil	Brasil	Miro	Dom	Zero	Wien	Ubu	Titanic
Codierung	00000	00001	0001	001	010	011	10	11



1. Aufgabe

Gast-stätte	Krokodil	Brasil	Miro	Dom	Zero	Wien	Ubu	Titanic
Codierung	00000	00001	0001	001	010	011	10	11



1. Aufgabe

1.4 Wie hoch ist die mittlere Codewortlänge und welchen Wert weist das theoretische Minimum auf?

Gast-stätte	Titanic	Ubu	Wien	Zero	Dom	Miro	Brasil	Krokodil
p	0,231	0,231	0,192	0,115	0,077	0,077	0,0385	0,0385
Code	11	10	011	010	001	0001	00001	00000
m	2	2	3	3	3	4	5	5

1. Aufgabe

Gast-stätte	Titanic	Ubu	Wien	Zero	Dom	Miro	Brasil	Krokodil
p	0,231	0,231	0,192	0,115	0,077	0,077	0,0385	0,0385
m	2	2	3	3	3	4	5	5

mittlere Codewortlänge:

$$\overline{m} = \sum_{i=1}^{n} p(x_i) \cdot m(x_i) = 0.231 \cdot 2 + \ldots + 0.0385 \cdot 5 \approx 2{,}77 \text{ bit}$$

das theoretische Minimum (Durchschnittlicher Informationsgehalt) :

$$H = \sum_{i=1}^{n} p(x_i) \cdot \mathrm{ld} \frac{1}{p(x_i)} = 0.231 * ld \frac{1}{0.231} + \ldots + 0.0385 * ld \frac{1}{0.0385} \approx 2{,}72 \text{ bit}$$

2. Aufgabe

Ergänzen Sie folgende Tabelle durch Konvertierung der angegebenen Zahlen in die jeweiligen Zahlensysteme.

dual	oktal	dezimal	hexadezimal
1011 1001			
1101 0110			
	363		
	1021		
		317	
		1150	
			ED3
			C8E

2. Aufgabe

Dez	Bin	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

Dez	Bin	Hex
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

2. Aufgabe

Beispiele:

$$\underbrace{(11}\underbrace{010}\underbrace{110)}_{B}$$

$$= (3 \quad 2 \quad 6)_o$$

$$1021_o$$

$$=(\underbrace{001}\ \underbrace{000}\ \underbrace{010}\ \underbrace{001})_B$$

$$=(2 \quad 1 \quad 1)_H$$

$$=(2*16^2+1*16^1+1*16^0)_D$$

$$= 529\ D$$

$$(\underbrace{1101}\underbrace{0110})_B$$

$$= (D \quad 6)_H$$

Wandlung der Dezimalzahl in andere Systemen

Dezimalzahl: N_D

10^n .... 10^4 10^3 10^2 10^1 10^0

$\mathbf{N_D}$ **mod** $\mathbf{R_x}$ (Rest der Division N_D / R_X)

$\mathbf{N_D}$ **div** $\mathbf{R_x}$ = $\mathbf{N_{D1}}$ (Vorkommastellen der Division N_D/R_x)

$\mathbf{N_{D1}}$ **mod** $\mathbf{R_x}$

$\mathbf{N_{D2}}$ **mod** $\mathbf{R_x}$

⋮

$\mathbf{N_{Dm}}$ **mod** $\mathbf{R_x}$

Zahl in beliebigem Zahlensystem: $\mathbf{N_{irgendwas}}$

$R_x{}^m$ .... $R_x{}^4$ $R_x{}^3$ $R_x{}^2$ $R_x{}^1$ $R_x{}^0$

Basis $\mathbf{R_x}$ des Zahlensystems: $\mathbf{N_{irgendwas}}$

Wandlung der Dezimalzahl in andere Systemen

Dezimalzahl: $N_D = 2004$

2	0	0	4
10^3	10^2	10^1	10^0

Beispiel:
Wandlung der Dezimalzahl 2004 in eine Hexadezimalzahl.

2004 mod 16 = 4

2004 div 16 = 125

125 mod 16 = 13

125 div 16 = 7

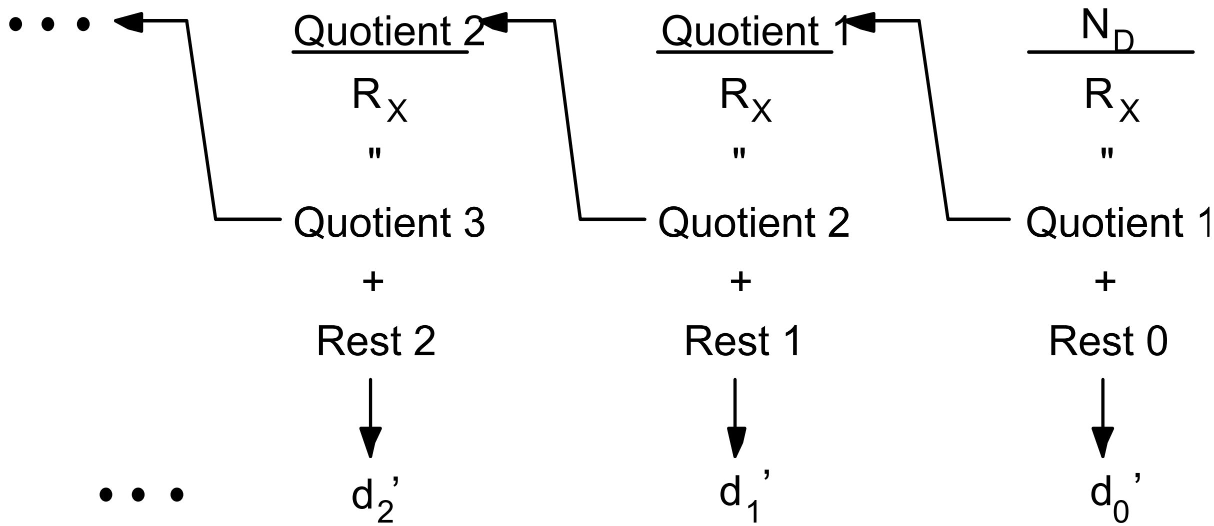
7 mod 16 = 7

Hexadezimalzahl: N_H

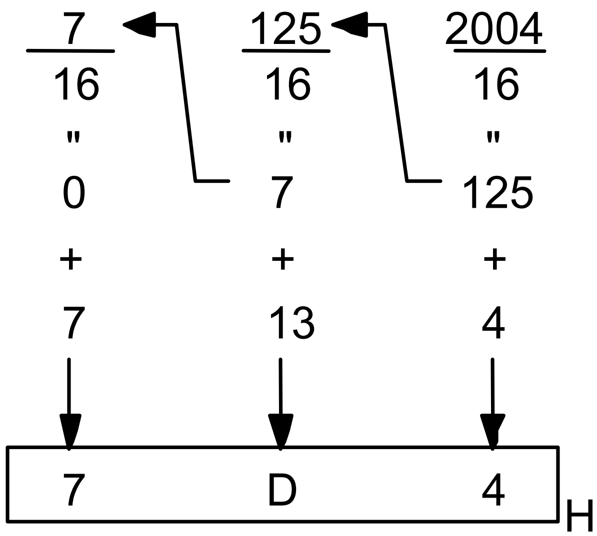
7	D	4
16^2	16^1	16^0

KIT – Universität des Landes Baden-Württemberg und nationales Forschungszentrum in der Helmholtz-Gemeinschaft
Institut für Technik der Informationsverarbeitung

Wandlung der Dezimalzahl in andere Systemen



Beispiel:
Wandlung der Dezimalzahl 2004 in eine Hexadezimalzahl.



2. Aufgabe

1150_D :

1150 / 8 = 143	Rest 6
143 / 8 = 17	Rest 7
17 / 8 = 2	Rest 1
2 / 8 = 0	Rest 2

1150_D
$= 2176_O$
$= 010\ 001\ 111\ 110_B$
$= 47E_H$

2. Aufgabe

dual	oktal	dezimal	hexadezimal
1011 1001	271	185	B9
1101 0110	326	214	D6
1111 0011	363	243	F3
0010 0001 0001	1021	529	211
0001 0011 1101	475	317	13D
0100 0111 1110	2176	1150	47E
1110 1101 0011	7323	3795	ED3
1100 1000 1110	6216	3214	C8E

3. Aufgabe

Gegeben sei ein nicht fehlertolerantes Kommunikationssystem, das in der Lage ist, Datenworte unterschiedlicher Länge zu erzeugen. Dieses soll nun dahingehend erweitert werden, dass bis zu drei Fehler erkannt und korrigiert werden können.

3.1 Bestimmen Sie anhand der unten gegebenen Tabelle die Mindestanzahl von Prüfbits zur Korrektur:

a) eines Fehlers für Datenwörter mit einer Breite von:
3Bit, 9Bit, 14Bit

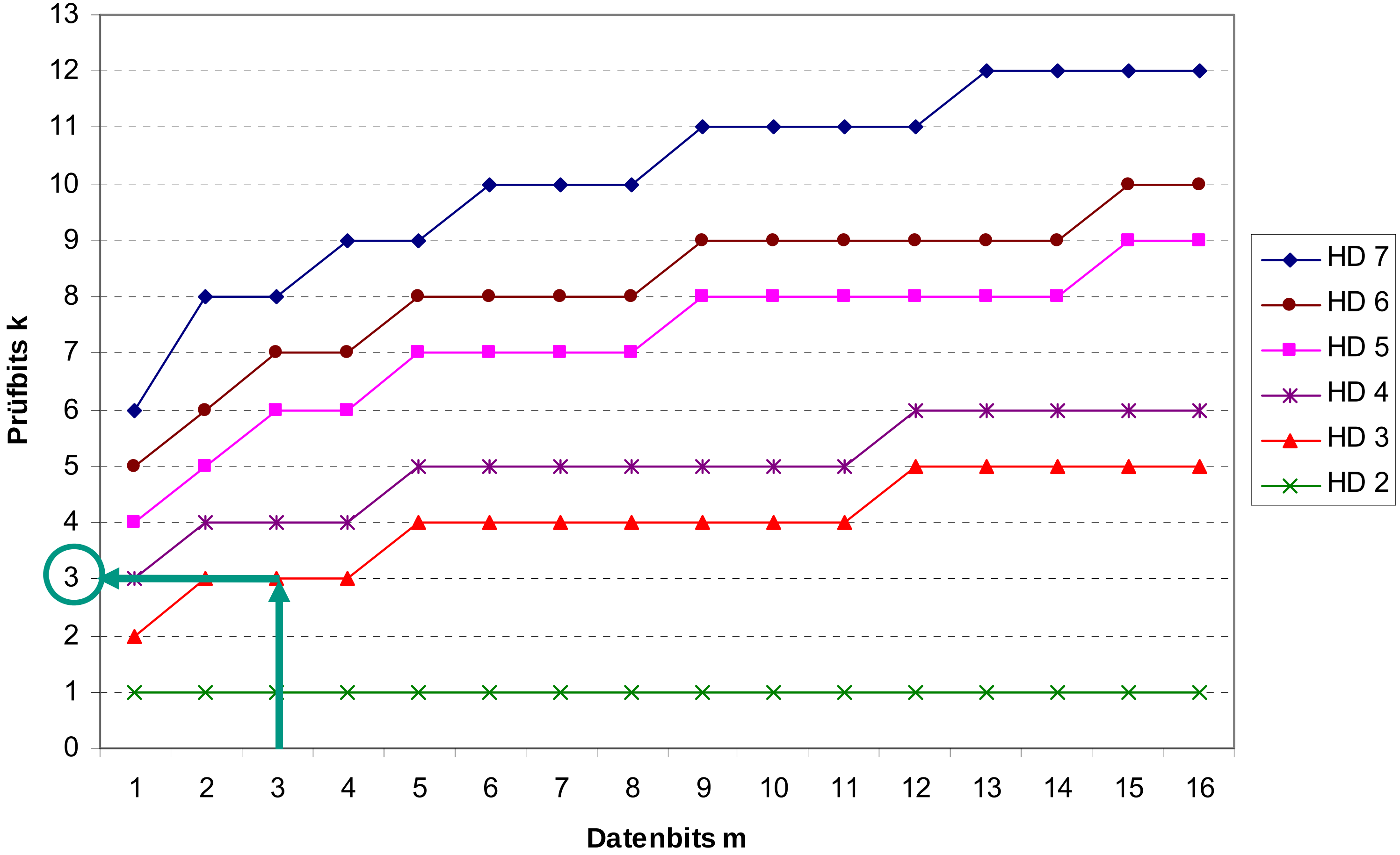
Zur Korrektur eines Fehlers wird eine minimale Hammingdistanz HD_{min}

$$HD_{min} = 2e+1 = 2*1+1 = 3$$

benötigt

3. Aufgabe

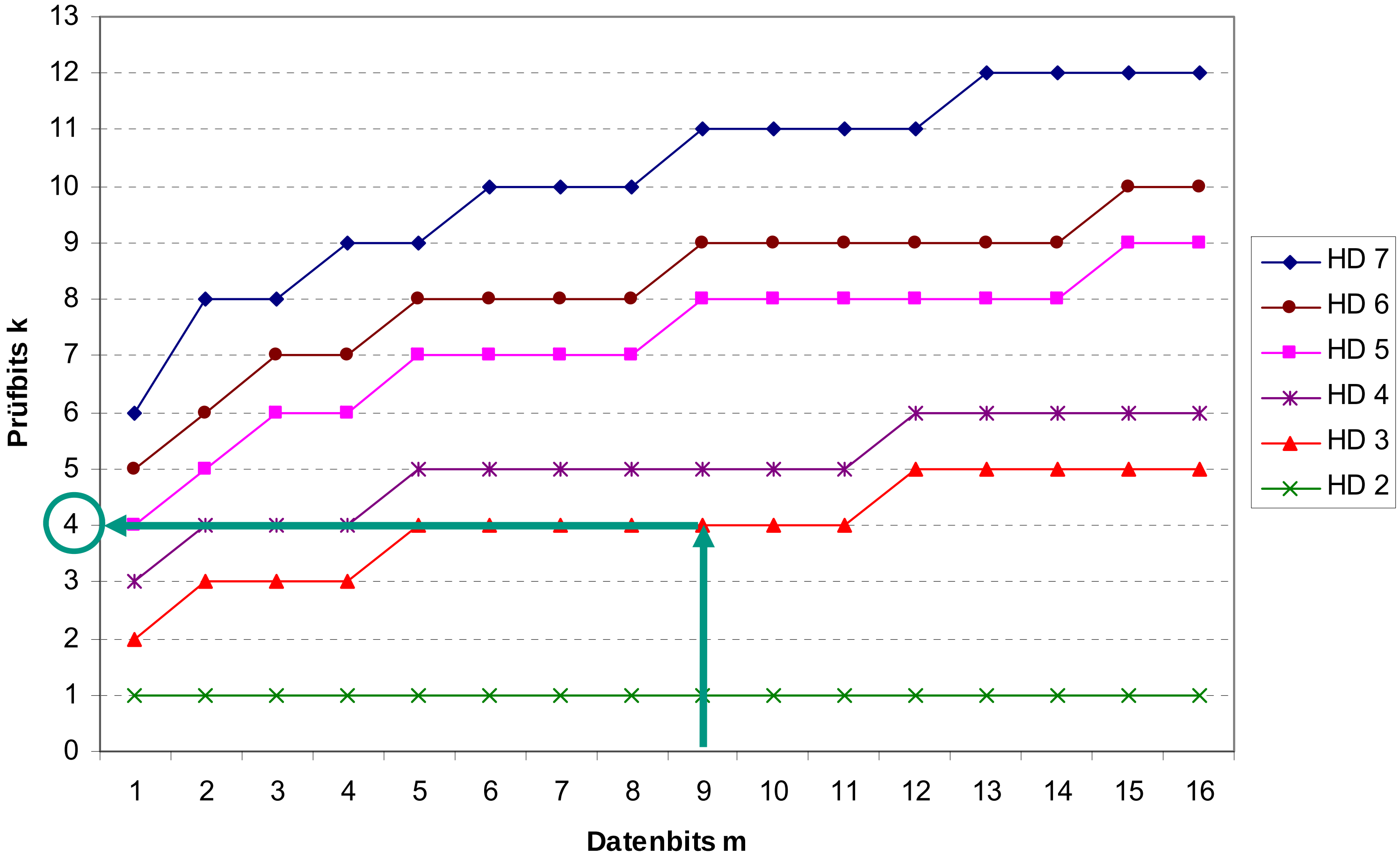
$$HD_{min} = 2e+1 = 2*1+1 = 3;\ m = 3\ \text{Bit}$$



KIT – Universität des Landes Baden-Württemberg und nationales Forschungszentrum in der Helmholtz-Gemeinschaft
Institut für Technik der Informationsverarbeitung

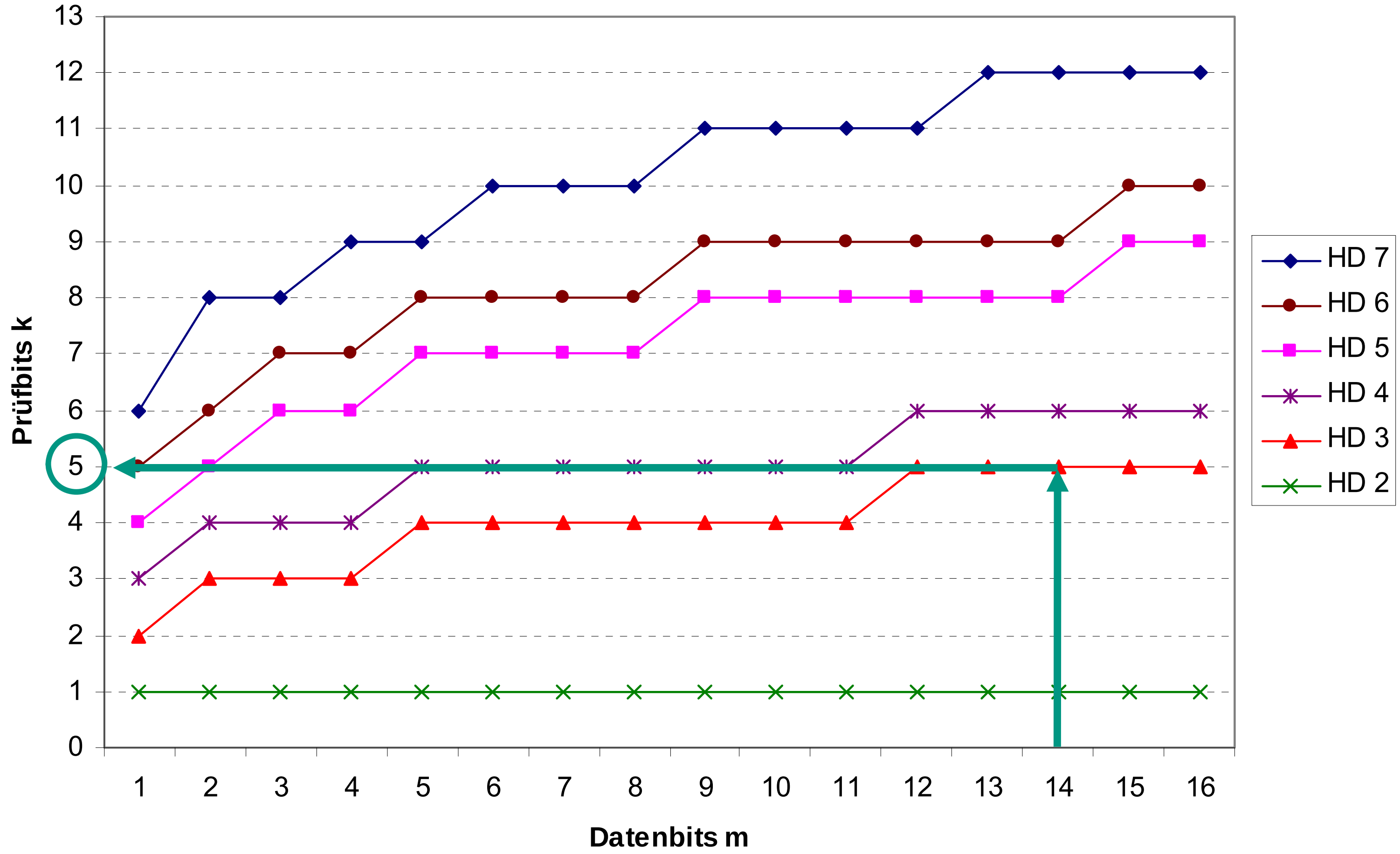
3. Aufgabe

$HD_{min} = 2e+1 = 2*1+1 = 3; m = 9$ Bit



3. Aufgabe

$$HD_{min} = 2e+1 = 2*1+1 = 3; \; m = 14 \text{ Bit}$$



3. Aufgabe

Gegeben sei ein nicht fehlertolerantes Kommunikationssystem, das in der Lage ist, Datenworte unterschiedlicher Länge zu erzeugen. Dieses soll nun dahingehend er-weitert werden, dass bis zu drei Fehler erkannt und korrigiert werden können.

3.1 Bestimmen Sie anhand der unten gegebenen Tabelle die Mindestanzahl von Prüfbits zur Korrektur:

b) von zwei Fehlern für Datenwörter mit einer Breite von: 4Bit, 8Bit, 15Bit

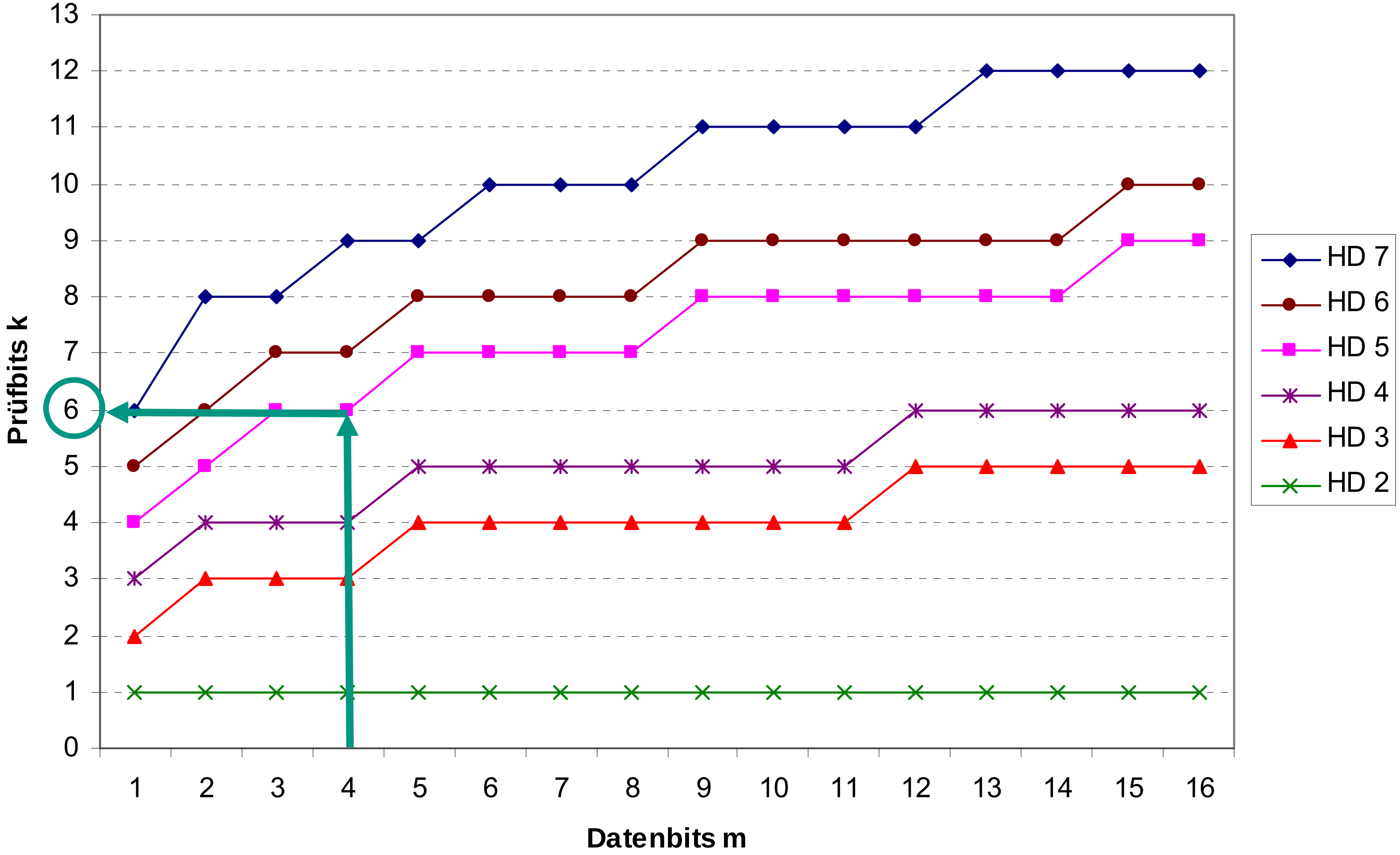
Zur Korrektur zweier Fehler wird eine minimale Hammingdistanz HD_{min}

$$HD_{min} = 2e+1 = 2*2+1 = 5$$

benötigt

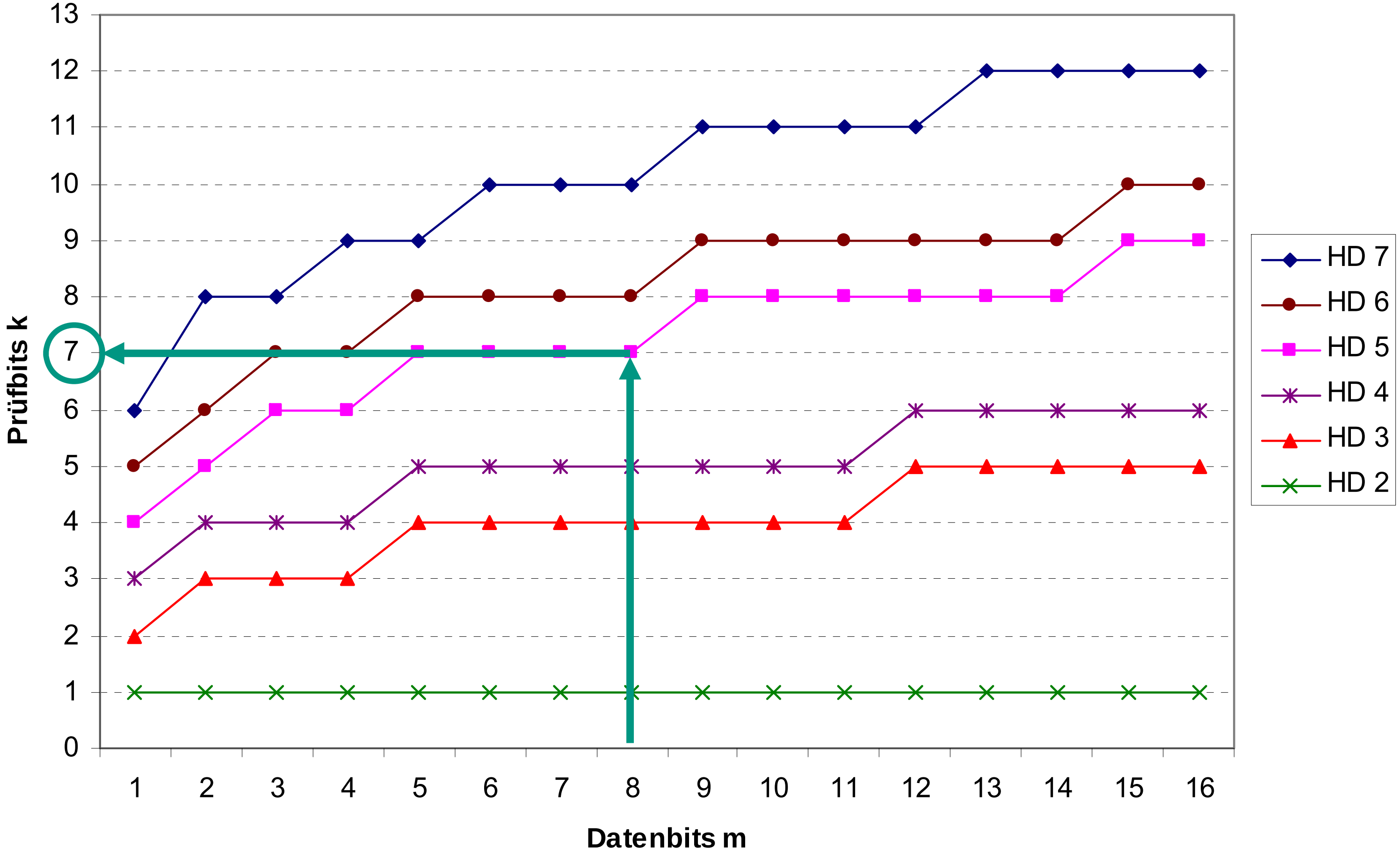
3. Aufgabe

HD_{min} = 2e+1= 2*2+1 = 5; m = 4 Bit



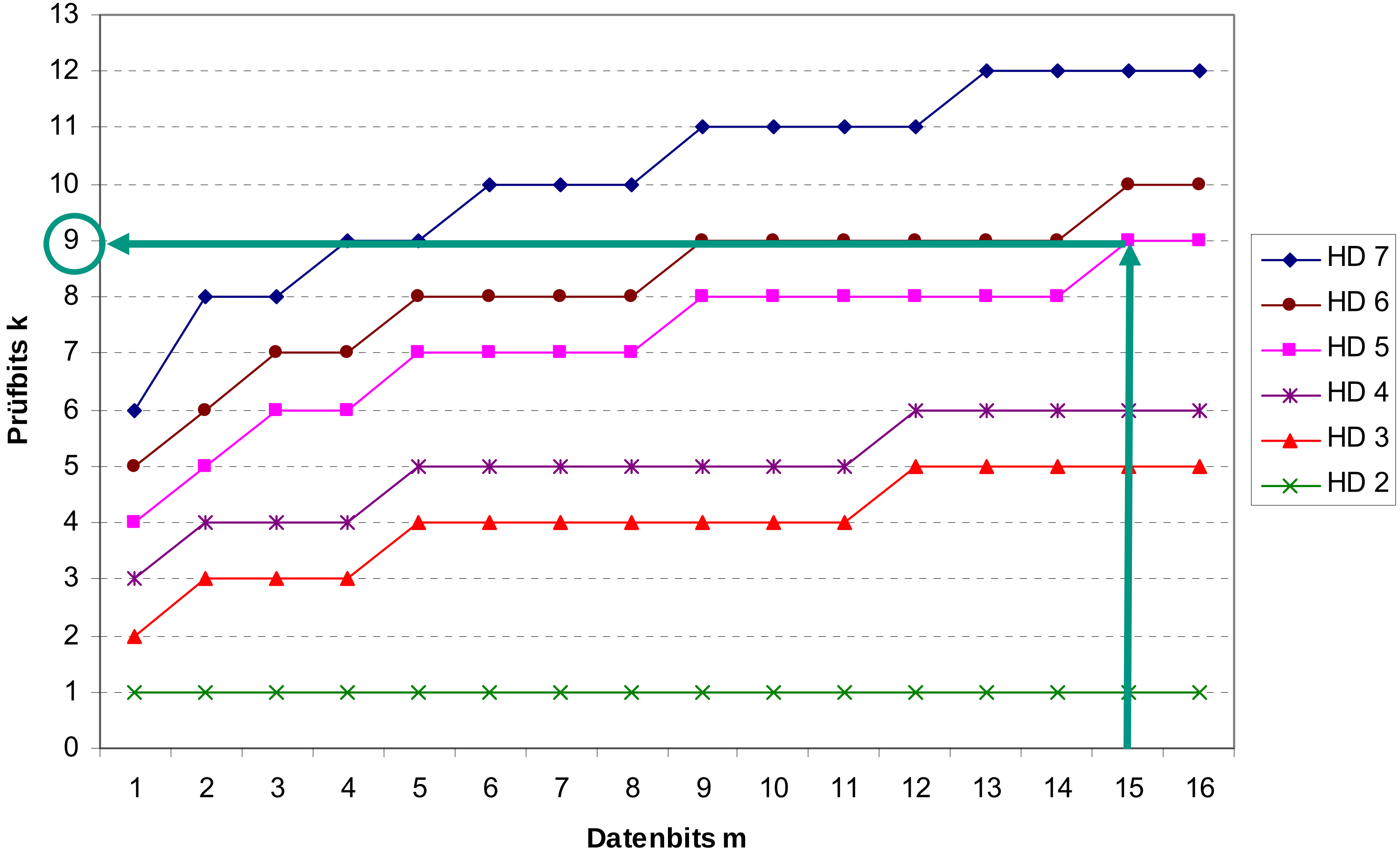
3. Aufgabe

$$HD_{min} = 2e+1 = 2*2+1 = 5;\ m = 8\ \text{Bit}$$



3. Aufgabe

HD_{min} = 2e+1= 2*2+1 = 5; m = 15 Bit



3. Aufgabe

Gegeben sei ein nicht fehlertolerantes Kommunikationssystem, das in der Lage ist, Datenworte unterschiedlicher Länge zu erzeugen. Dieses soll nun dahingehend erweitert werden, dass bis zu drei Fehler erkannt und korrigiert werden können.

3.1 Bestimmen Sie anhand der unten gegebenen Tabelle die Mindestanzahl von Prüfbits zur Korrektur:

c) von drei Fehlern für Datenwörter mit einer Breite von:
5Bit, 9Bit, 13Bit

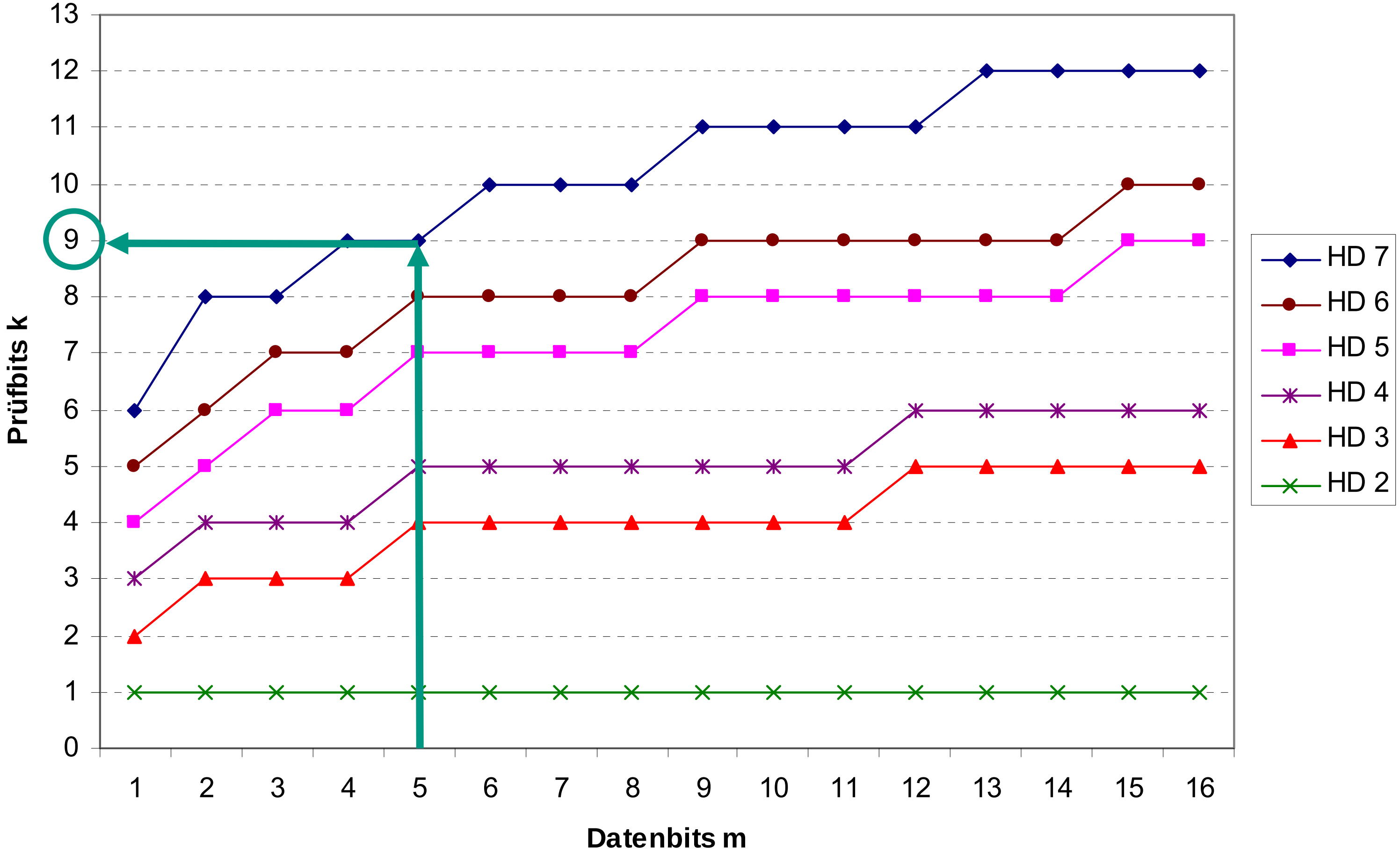
Zur Korrektur von drei Fehler wird
eine minimale Hammingdistanz HD_{min}

$$HD_{min} = 2e+1 = 2*3+1 = 7$$

benötigt

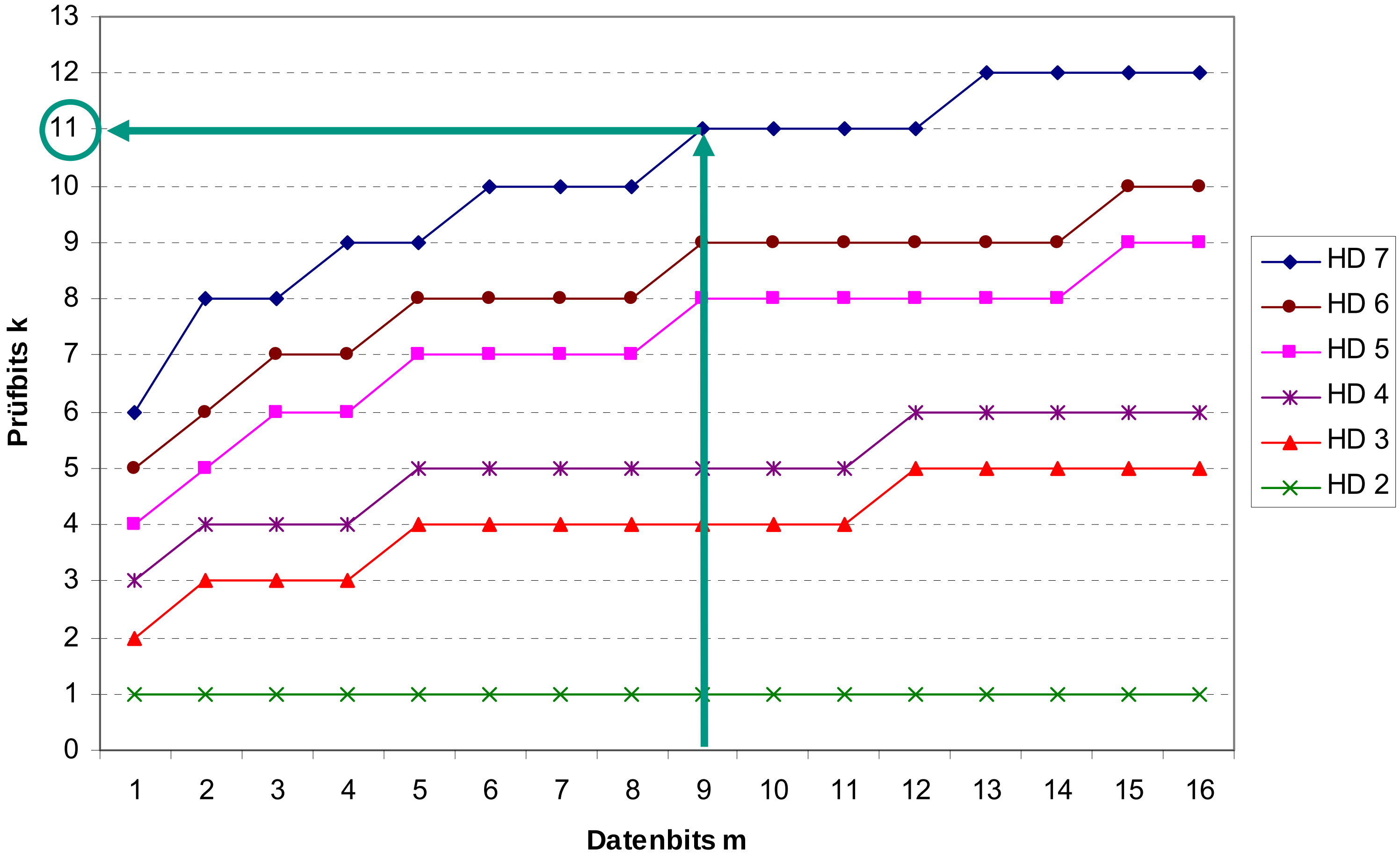
3. Aufgabe

$$HD_{min} = 2e+1 = 2*3+1 = 7; \; m = 5 \text{ Bit}$$



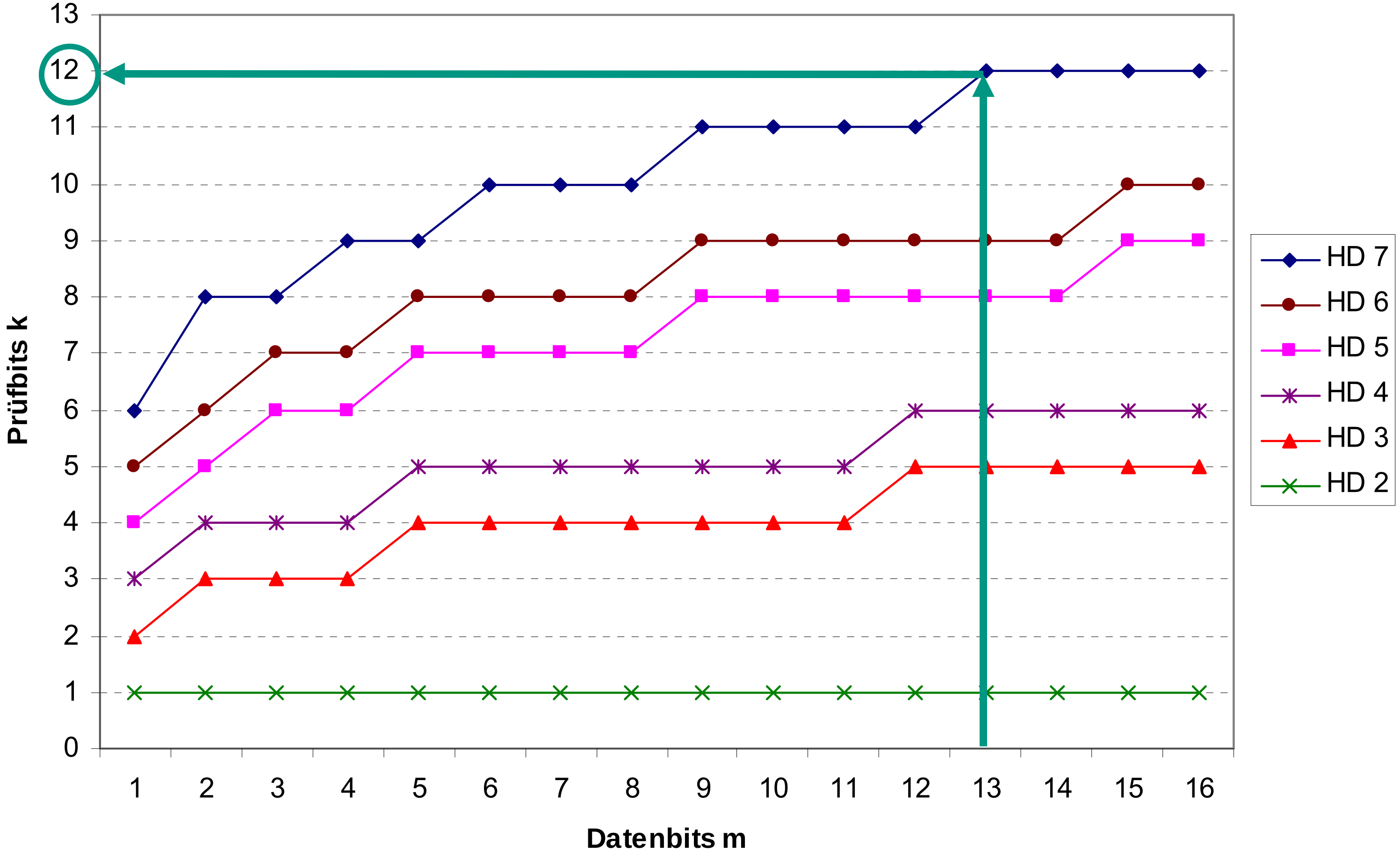
3. Aufgabe

$$HD_{min} = 2e+1 = 2*3+1 = 7; \; m = 9 \text{ Bit}$$



3. Aufgabe

HD_{min} = 2e+1= 2*3+1 = 7; m = 13 Bit



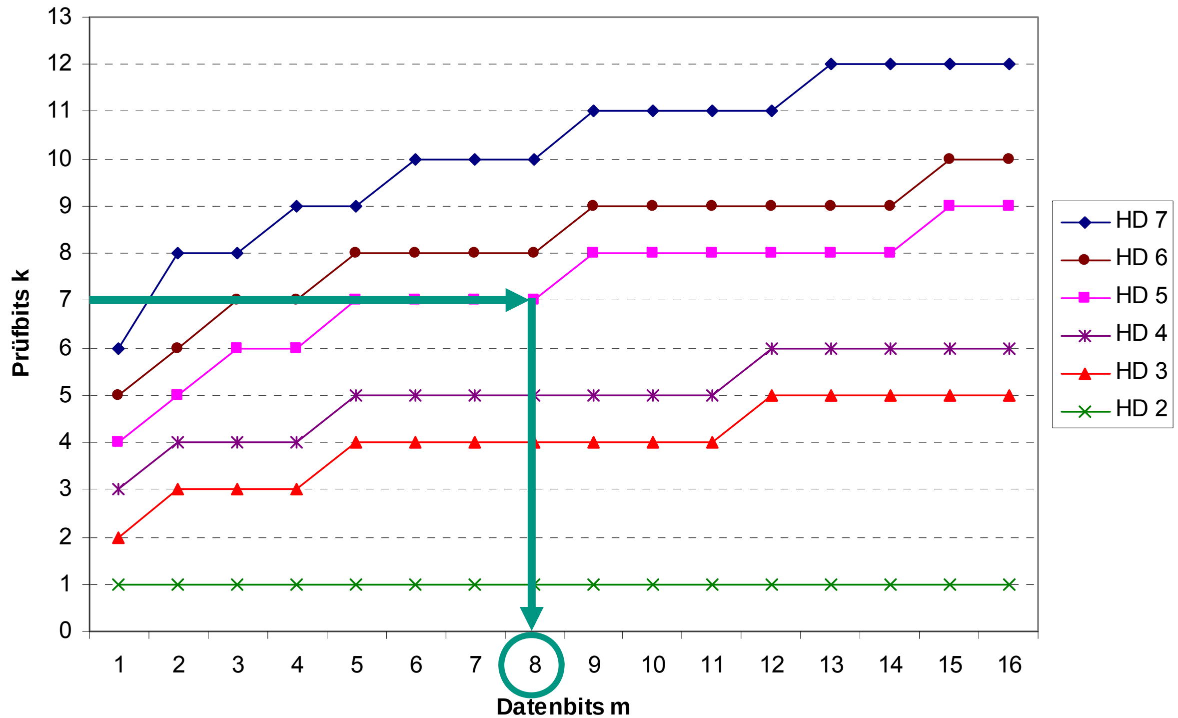
3. Aufgabe

Gegeben sei ein nicht fehlertolerantes Kommunikationssystem, das in der Lage ist, Datenworte unterschiedlicher Länge zu erzeugen. Dieses soll nun dahingehend er-weitert werden, dass bis zu drei Fehler erkannt und korrigiert werden können.

3.2 Bestimmen Sie die maximale Anzahl an Datenbits, die mit Hilfe von 7 Prüfbits gegen das Auftreten von maximal zwei Fehlern gesichert werden können.

HD_{min} zur Korrektur zweier Fehler

$$HD_{min} = 2e+1 = 5$$



3. Aufgabe

Gegeben sei ein nicht fehlertolerantes Kommunikationssystem, das in der Lage ist, Datenworte unterschiedlicher Länge zu erzeugen. Dieses soll nun dahingehend erweitert werden, dass bis zu drei Fehler erkannt und korrigiert werden können.

3.3 Im Folgenden soll nun davon ausgegangen werden, dass das Kommunikationssystem in der Lage ist, 16 unterschiedliche Datenworte zu übertragen. Es soll nun dahingehend erweitert werden, dass es mittels eines Hamming-Codes einen Fehler korrigieren kann.

- Bestimmen Sie zunächst die minimale Anzahl der benötigen Datenbits, sowie die Anzahl der erforderlichen Prüfbits, um die gewünschte Fehlertoleranz zu erzielen.

$$\Rightarrow m = \lceil \mathrm{ld}\ 16 \rceil = 4 \text{ Bits}$$

$$1 \text{ Fehler-Korrektur} \Rightarrow HD_{min} = 3$$

$$\Rightarrow k = 3 \text{ Bits}$$

3. Aufgabe

Gegeben sei ein nicht fehlertolerantes Kommunikationssystem, das in der Lage ist, Datenworte unterschiedlicher Länge zu erzeugen. Dieses soll nun dahingehend erweitert werden, dass bis zu drei Fehler erkannt und korrigiert werden können.

3.3 Im Folgenden soll nun davon ausgegangen werden, dass das Kommunikationssystem in der Lage ist, 16 unterschiedliche Datenworte zu übertragen. Es soll nun dahingehend erweitert werden, dass es mittels eines Hamming-Codes einen Fehler korrigieren kann.

- Konstruieren Sie nun den Hamming-Code auf Basis des in der Vorlesung eingeführten tabellenbasierten Verfahrens. Beachten Sie, dass in diesem Fall die Prüfbits auf ungerade Parität zu ergänzen sind.

Anmerkung zur letzten Übung : Hamming-Codes

Systematik: Konstruktion von Hamming-Codes

Zuordnung der geprüften Stellen zu den Prüfstellen:

i. Prüfstelle y_i überprüft **alle Binärstellen (können Informationsstellen m als auch Prüfstellen k sein!),** die in der **i. Stelle (im i. Bit)** der **Dualzahlendarstellungen** der entsprechenden **Binärstellenpositionen (“duale Kennzahlen“)** eine **“1“** aufweisen

Prüfstellen	der Prüfstelle zugeordnete Binärstelle	geprüfte Binärstellen
y1	1	1, 3, 5, 7, 9, 11, 13, 15, 17, ...
y2	2	2, 3, 6, 7, 10, 11, 14, 15, 18, ...
y3	3	4, 5, 6, 7, 12, 13, 14, 15, 20, ...
y4	4	8, 9, 10, 11, 12, 13, 14, 15, 24, ...
.	.	...
.	.	
.	.	

Konstruktion von Hamming-Codes

Konstruktion der Tabelle zum Generieren des Hamming-Codes

Aufbau eines 1-F-korrigierbaren Codes:

- Einfachfehler sind korrigierbar: **HD_{min} = 3**
- Anzahl der Informationsstellen m = 4
  - ➔ Anzahl der Prüfstellen k = 3

- **Zuordnung der geprüften Stellen zu den Prüfstellen:**

➢ **1. Prüfstelle y_1** überprüft **alle Binärstellen,** die in der **1. Stelle (im 1. Bit)** der **Dualzahlendarstellungen** der entsprechenden **Binärstellenpositionen (“duale Kennzahlen“)** eine **“1“** aufweisen

1. Schritt	lfd. Nr. duale Kennzahl	1 001	2 010	3 011	4 100	5 101	6 110	7 111
2. Schritt	1. Stellenbelegung Bestimmung von y_1	y_1		x_1		x_2		x_3
3. Schritt	2. Stellenbelegung Bestimmung von y_2							
4. Schritt	3. Stellenbelegung Bestimmung von y_3							

KIT – Universität des Landes Baden-Württemberg und nationales Forschungszentrum in der Helmholtz-Gemeinschaft
Institut für Technik der Informationsverarbeitung

Konstruktion von Hamming-Codes

- **Zuordnung der geprüften Stellen zu den Prüfstellen:**

➢ **2. Prüfstelle y_2** überprüft **alle Binärstellen,** die in der **2. Stelle (im 2. Bit)** der **Dualzahlendarstellungen** der entsprechenden **Binärstellenpositionen („duale Kennzahlen")** eine **„1"** aufweisen

1. Schritt	lfd. Nr. duale Kennzahl	1 001	2 010	3 011	4 100	5 101	6 110	7 111
2. Schritt	1. Stellenbelegung Bestimmung von y_1	y_1		x_1		x_2		x_3
3. Schritt	2. Stellenbelegung Bestimmung von y_2		y_2	x_1			x_4	x_3
4. Schritt	3. Stellenbelegung Bestimmung von y_3							

Konstruktion von Hamming-Codes

- **Zuordnung der geprüften Stellen zu den Prüfstellen:**

➢ **3. Prüfstelle y_3** überprüft **alle Binärstellen,** die in der **3. Stelle (im 3. Bit)** der **Dualzahlendarstellungen** der entsprechenden **Binärstellenpositionen („duale Kennzahlen")** eine **„1"** aufweisen

1. Schritt	lfd. Nr. duale Kennzahl	1 001	2 010	3 011	4 100	5 101	6 110	7 111
2. Schritt	1. Stellenbelegung Bestimmung von y_1	y_1		x_1		x_2		x_3
3. Schritt	2. Stellenbelegung Bestimmung von y_2		y_2	x_1			x_4	x_3
4. Schritt	3. Stellenbelegung Bestimmung von y_3				y_3	x_2	x_4	x_3

Konstruktion von Hamming-Codes

- Entwicklung eines 1 F - korrigierbaren Codes mit
  - m = 4: Informationsstellen
  - k = 3: Prüfstellen
  - Ergänzung auf ungerade Parität

Codewort (CW)		x1	x2	x3	x4	y1	y2	y3	Dezimalzahl
1.	CW	0	0	0	0				0
2.	CW	0	0	0	1				1
3.	CW	0	0	1	0				2
4.	CW	0	0	1	1				3
5.	CW	0	1	0	0				4
6.	CW	0	1	0	1				5
7.	CW	0	1	1	0				6
8.	CW	0	1	1	1				7
9.	CW	1	0	0	0				8
10.	CW	1	0	0	1				9
11.	CW	1	0	1	0				10
12.	CW	1	0	1	1				11
13.	CW	1	1	0	0				12
14.	CW	1	1	0	1				13
15.	CW	1	1	1	0				14
16.	CW	1	1	1	1				15

Konstruktion von Hamming-Codes

- Entwicklung des 1. Codewortes: <$x_1x_2x_3x_4$> = 0000

1. Schritt	lfd. Nr. duale Kennzahl	1 001	2 010	3 011	4 100	5 101	6 110	7 111
2. Schritt	1. Stellenbelegung Bestimmung von y_1	y_1 1		x_1 0		x_2 0		x_3 0
3. Schritt	2. Stellenbelegung Bestimmung von y_2		y_2 1	x_1 0			x_4 0	x_3 0
4. Schritt	3. Stellenbelegung Bestimmung von y_3				y_3 1	x_2 0	x_4 0	x_3 0

- Entwicklung des 2. Codewortes: <$x_1x_2x_3x_4$> = 0001

1. Schritt	lfd. Nr. duale Kennzahl	1 001	2 010	3 011	4 100	5 101	6 110	7 111
2. Schritt	1. Stellenbelegung Bestimmung von y_1	y_1 1		x_1 0		x_2 0		x_3 0
3. Schritt	2. Stellenbelegung Bestimmung von y_2		y_2 0	x_1 0			x_4 1	x_3 0
4. Schritt	3. Stellenbelegung Bestimmung von y_3				y_3 0	x_2 0	x_4 1	x_3 0

Konstruktion von Hamming-Codes

Teilweise vervollständigte Codewort Tabelle

Codewort (CW)		x1	x2	x3	x4	y1	y2	y3	Dezimalzahl
1.	CW	0	0	0	0	1	1	1	0
2.	CW	0	0	0	1	1	0	0	1
3.	CW	0	0	1	0				2
4.	CW	0	0	1	1				3
5.	CW	0	1	0	0				4
6.	CW	0	1	0	1				5
7.	CW	0	1	1	0				6
8.	CW	0	1	1	1				7
9.	CW	1	0	0	0				8
10.	CW	1	0	0	1				9
11.	CW	1	0	1	0				10
12.	CW	1	0	1	1				11
13.	CW	1	1	0	0				12
14.	CW	1	1	0	1				13
15.	CW	1	1	1	0				14
16.	CW	1	1	1	1				15

Konstruktion von Hamming-Codes

- Entwicklung des 3. Codewortes: $<x_1x_2x_3x_4> = 0010$

1. Schritt	lfd. Nr. duale Kennzahl	1 001	2 010	3 011	4 100	5 101	6 110	7 111
2. Schritt	1. Stellenbelegung Bestimmung von y_1	y_1 0		x_1 0		x_2 0		x_3 1
3. Schritt	2. Stellenbelegung Bestimmung von y_2		y_2 0	x_1 0			x_4 0	x_3 1
4. Schritt	3. Stellenbelegung Bestimmung von y_3				y_3 0	x_2 0	x_4 0	x_3 1

- Entwicklung des 4. Codewortes: $<x_1x_2x_3x_4> = 0011$

1. Schritt	lfd. Nr. duale Kennzahl	1 001	2 010	3 011	4 100	5 101	6 110	7 111
2. Schritt	1. Stellenbelegung Bestimmung von y_1	y_1 0		x_1 0		x_2 0		x_3 1
3. Schritt	2. Stellenbelegung Bestimmung von y_2		y_2 1	x_1 0			x_4 1	x_3 1
4. Schritt	3. Stellenbelegung Bestimmung von y_3				y_3 1	x_2 0	x_4 1	x_3 1

Konstruktion von Hamming-Codes

Teilweise vervollständigte Codewort Tabelle

Codewort (CW)		x1	x2	x3	x4	y1	y2	y3	Dezimalzahl
1.	CW	0	0	0	0	1	1	1	0
2.	CW	0	0	0	1	1	0	0	1
3.	CW	0	0	1	0	0	0	0	2
4.	CW	0	0	1	1	0	1	1	3
5.	CW	0	1	0	0				4
6.	CW	0	1	0	1				5
7.	CW	0	1	1	0				6
8.	CW	0	1	1	1				7
9.	CW	1	0	0	0				8
10.	CW	1	0	0	1				9
11.	CW	1	0	1	0				10
12.	CW	1	0	1	1				11
13.	CW	1	1	0	0				12
14.	CW	1	1	0	1				13
15.	CW	1	1	1	0				14
16.	CW	1	1	1	1				15

Konstruktion von Hamming-Codes

Vervollständigte Codewort Tabelle

Codewort (CW)		x1	x2	x3	x4	y1	y2	y3	Dezimalzahl
1.	CW	0	0	0	0	1	1	1	0
2.	CW	0	0	0	1	1	0	0	1
3.	CW	0	0	1	0	0	0	0	2
4.	CW	0	0	1	1	0	1	1	3
5.	CW	0	1	0	0	0	1	0	4
6.	CW	0	1	0	1	0	0	1	5
7.	CW	0	1	1	0	1	0	1	6
8.	CW	0	1	1	1	1	1	0	7
9.	CW	1	0	0	0	0	0	1	8
10.	CW	1	0	0	1	0	1	0	9
11.	CW	1	0	1	0	1	1	0	10
12.	CW	1	0	1	1	1	0	1	11
13.	CW	1	1	0	0	1	0	0	12
14.	CW	1	1	0	1	1	1	1	13
15.	CW	1	1	1	0	0	1	1	14
16.	CW	1	1	1	1	0	0	0	15

3. Aufgabe

Gegeben sei ein nicht fehlertolerantes Kommunikationssystem, das in der Lage ist, Datenworte unterschiedlicher Länge zu erzeugen. Dieses soll nun dahingehend erweitert werden, dass bis zu drei Fehler erkannt und korrigiert werden können.

3.4 Einige der in Aufgabe 3.3 erzeugten Wörter sollen nun übertragen werden. Nachfolgend ist der am Empfänger aufgezeichnete Bitstrom angegeben.

01001010111111111110010011

Ermitteln Sie die enthaltenen Fehler und korrigieren Sie den Bitstrom entsprechend. Das LSB eines jeden übertragenen Wortes befindet sich dabei äußerst links. Der empfangene Stream setzt sich also wie folgt zusammen:

$$\underbrace{\text{Bit 1 ... Bit n}}_{\text{1. Wort}},\underbrace{\text{Bit 1 ... Bit n}}_{\text{2. Wort}},...$$

3. Aufgabe

Am Empfänger aufgezeichneter Bitstrom:

0100101011111111111110010011

m = 4 Bits, k = 3 Bits $\Rightarrow$ Wortlänge: m+k = 7 Bits

Zerlegen des Bitstroms:

1. Wort: 0100101
2. Wort: 0111111
3. Wort: 1111111
4. Wort: 0010011

Anordnung der übertragenen Bits:

1.Bit	2.Bit	3.Bit	4.Bit	5.Bit	6.Bit	7.bit
Y_1	Y_2	X_1	Y_3	X_2	X_4	X_3

3. Aufgabe

Möglichkeiten zur Fehlerkorrektur:

1. Ermitteln der Hammingdistanz der empfangenen Bitfolge zu jedem Eintrag in der Codewort-Tabelle
   - Für HD = 0 ist eine Übereinstimmung gefunden, kein Fehler liegt vor
   - Für HD = 1 liegt ein Fehler vor, Korrektur möglich
   - Für HD > 1 sind mehrer Fehler aufgetreten, kein Korrektur möglich

   Nachteil: Langwieriges Verfahren

2. Bestimmen des Fehlers mit Hilfe eines Systematischen Verfahrens:
   - Duales Kennzahlverfahren

3. Aufgabe

Beispiel für systematische Korrektur von: <$y_1,y_2,x_1,y_3,x_2,x_4,x_3$>= 0100101 für die Annahme, dass maximal ein Fehler aufgetreten ist!

Vorgehen: Markieren der Zeilen, bei der die Paritätsbedingung nicht eingehalten wird. Das Bit, das dabei in <u>allen</u> markierten Zeilen enthalten ist, beinhaltet den Fehler.

1. Schritt	lfd. Nr. duale Kennzahl	1 001	2 010	3 011	4 100	5 101	6 110	7 111
2. Schritt	1. Stellenbelegung Bestimmung von y_1	y_1 0		x_1 0		x_2 1		x_3 1
3. Schritt	2. Stellenbelegung Bestimmung von y_2		y_2 1	x_1 0			x_4 0	x_3 1
4. Schritt	3. Stellenbelegung Bestimmung von y_3				y_3 0	x_2 1	x_4 0	x_3 1

$\Rightarrow$ Bit 7 ist das einzige Bit, das in allen fehlerhaften Zeilen enthalten ist. Bit 7 ist also fehlerhaft. Das korrekte Codewort lautet somit : 0100100

KIT – Universität des Landes Baden-Württemberg und nationales Forschungszentrum in der Helmholtz-Gemeinschaft
Institut für Technik der Informationsverarbeitung

3. Aufgabe

Korrektur von: <$y_1,y_2,x_1,y_3,x_2,x_4,x_3$>= 0111111
für die Annahme, dass maximal ein Fehler aufgetreten ist!

Vorgehen: Fehlerhafte Prüfbits werden mit einer 1 markiert

1. Schritt	lfd. Nr. duale Kennzahl	1 001	2 010	3 011	4 100	5 101	6 110	7 111	Fehlerhaftes Prüfbit
2. Schritt	1. Stellenbelegung Bestimmung von y_1	y_1 0		x_1 1		x_2 1		x_3 1	0
3. Schritt	2. Stellenbelegung Bestimmung von y_2		y_2 1	x_1 1			x_4 1	x_3 1	1
4. Schritt	3. Stellenbelegung Bestimmung von y_3				y_3 1	x_2 1	x_4 1	x_3 1	1

Duale Kennzahl der fehlerhaften Stelle: 1 1 0

$\Rightarrow$ der Fehler befindet sich an der 6. Stelle im Datenwort
Fehlerhaftes Wort: 0111111; Korrigiertes Wort: 0111101

3. Aufgabe

- Korrektur der verbleibenden zwei Wörter:

  Empfangenes Wort: 1111111
  - Analyse des Wortes ergibt, dass ein Fehler an der 7. Stelle aufgetreten ist
  - Korrigiertes Wort: 1111110

  Empfangenes Wort: 0010011
  - Analyse des Wortes ergibt, dass ein Fehler an der 5. Stelle aufgetreten ist
  - Korrigiertes Wort: 0010111

4. Aufgabe

Bei einer Alarmanlage sollen alle Einbrüche und Einbruchsversuche in einem nichtflüchtigen Speicher protokolliert werden. Da bei diesem Baustein jede Speicherzelle nur 1x geschrieben werden kann, soll dieser möglichst gut ausgenutzt werden.

Aus den Statistiken der Polizei ist bekannt, dass Einbrüche durch eine Tür 4x häufiger sind als durch ein Bad- oder Küchenfenster. Weiterhin ist bekannt, dass Einbrüche durch eine Tür 8x häufiger sind als durch ein sonstiges Fenster im Erdgeschoss. Schließlich zeigt die Statistik, dass ein Einbruch durch ein Fenster im 1. Stockwerk 16x seltener auftritt, als ein Einbruch durch eine Tür.

4. Aufgabe

4.1 Berechnen Sie zuerst die Auftrittswahrscheinlichkeiten für einen Einbruchsort in einer Wohnung mit 1 Wohnungstür, 1 Balkontür, 1 Badfenster, 1 Küchenfenster, 1 Schlafzimmerfenster und 2 Wohnzimmerfenstern im Erdgeschoss sowie 2 Kinderzimmerfenster und 2 Atelierfenster im 1. Stock.

Ort	Häufigkeit	Wahrscheinlichkeit	Ermittelte Codierung
Wohnungstür			
Balkontür			
Badfenster			
Küchenfenster			
Schlazimmerfenster			
Wohnzimmerfenster 1			
Wohnzimmerfenster 2			
Kinderzimmerfenster 1			
Kinderzimmerfenster 2			
Atelierfenster 1			
Atelierfenster 2			

4. Aufgabe

Karlsruhe Institute of Technology

4.1 Berechnen Sie zuerst die Auftrittswahrscheinlichkeiten für einen Einbruchsort in einer Wohnung mit 1 Wohnungstür, 1 Balkontür, 1 Badfenster, 1 Küchenfenster, 1 Schlafzimmerfenster und 2 Wohnzimmerfenstern im Erdgeschoss sowie 2 Kinderzimmerfenster und 2 Atelierfenster im 1. Stock.

Ort	Häufigkeit	Wahrscheinlichkeit	Ermittelte Codierung
Wohnungstür	16		
Balkontür	16		
Badfenster	4		
Küchenfenster	4		
Schlazimmerfenster	2		
Wohnzimmerfenster 1	2		
Wohnzimmerfenster 2	2		
Kinderzimmerfenster 1	1		
Kinderzimmerfenster 2	1		
Atelierfenster 1	1		
Atelierfenster 2	1		

$$\sum = 50$$

4. Aufgabe

4.1 Berechnen Sie zuerst die Auftrittswahrscheinlichkeiten für einen Einbruchsort in einer Wohnung mit 1 Wohnungstür, 1 Balkontür, 1 Badfenster, 1 Küchenfenster, 1 Schlafzimmerfenster und 2 Wohnzimmerfenstern im Erdgeschoss sowie 2 Kinderzimmerfenster und 2 Atelierfenster im 1. Stock.

Ort	Häufigkeit	Wahrscheinlichkeit	Ermittelte Codierung	
Wohnungstür	16	$^{16}/_{50}$		WT:16/50
Balkontür	16	$^{16}/_{50}$		BT: 16/50
Badfenster	4	$^{4}/_{50}$		BF: 4/50
Küchenfenster	4	$^{4}/_{50}$		KF: 4/50
Schlazimmerfenster	2	$^{2}/_{50}$		SF: 2/50
Wohnzimmerfenster 1	2	$^{2}/_{50}$		WF1: 2/50
Wohnzimmerfenster 2	2	$^{2}/_{50}$		WF2: 2/50
Kinderzimmerfenster 1	1	$^{1}/_{50}$		KF1: 1/50
Kinderzimmerfenster 2	1	$^{1}/_{50}$		KF2: 1/50
Atelierfenster 1	1	$^{1}/_{50}$		AF1: 1/50
Atelierfenster 2	1	$^{1}/_{50}$		AF2: 1/50

$$\sum = 50$$

4. Aufgabe

Karlsruhe Institute of Technology

4.2 Erstellen Sie eine optimale Codierung für die möglichen Orte und tragen Sie Ihre Ergebnisse in die oben vorgegebene Tabelle ein. Nutzen Sie hierzu das Huffmanverfahren.

Ort	Häufigkeit	Wahrscheinlichkeit	Ermittelte Codierung
Wohnungstür	16	$^{16}/_{50}$	
Balkontür	16	$^{16}/_{50}$	
Badfenster	4	$^{4}/_{50}$	
Küchenfenster	4	$^{4}/_{50}$	
Schlazimmerfenster	2	$^{2}/_{50}$	
Wohnzimmerfenster 1	2	$^{2}/_{50}$	
Wohnzimmerfenster 2	2	$^{2}/_{50}$	
Kinderzimmerfenster 1	1	$^{1}/_{50}$	
Kinderzimmerfenster 2	1	$^{1}/_{50}$	
Atelierfenster 1	1	$^{1}/_{50}$	
Atelierfenster 2	1	$^{1}/_{50}$	

$$\sum = 50$$

WT:16/50
BT: 16/50
BF: 4/50
KF: 4/50
SF: 2/50
WF1: 2/50
WF2: 2/50
KF1: 1/50
KF2: 1/50
AF1: 1/50
AF2: 1/50

Huffman Kodierung

Huffman's Kodierungs-Algorithmus (HKA) (1)

- HKA konstruiert einen Baum T, der die Kodierung der einzelnen Zeichen repräsentiert
- Eine Queue Q wird dafür benutzt, um die zwei Zeichen mit der niedrigsten Auftrittswahrscheinlichkeit zu finden
- Diese zwei Zeichen werden zu einem Objekt verschmolzen, dessen Auftrittswahrscheinlichkeit die Summe der Einzelwahrscheinlichkeiten ist

Huffman Kodierung

Huffman's Kodierungs-Algorithmus (HKA) (2)

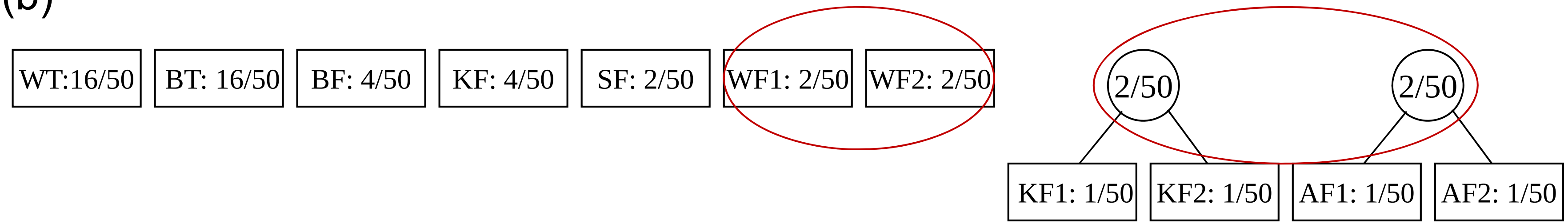
```
begin
    n ← |C|;              // C ist Menge der Elemente, n Anzahl der Elemente in C
    Q ← C;                // Q ist eine Kopie der Menge C
    for i ←1 to n-1 do
        z ← ALLOCATE-NODE();              // legt neuen Knoten z
        x ← left[z] ← EXTRACT-MIN(Q);     // findet das kleinste Element in Q
                                          // und hängt es links von z an
        y ← right[z] ← EXTRACT-MIN(Q);    // findet das kleinste Element in Q
                                          // und hängt es rechts von z an
        f[z] ← f[x] + f[y];               // z erhält die Auftrittswahrscheinlich,
                                          // die der Summer von x und y
                                          // entsprechen
        INSERT(z, Q);                     // z wird in Q eingefügt
    end;
    return EXTRACT-MIN(Q);                // das Ergebnis ist ein Huffman-Baum
end
```

4. Aufgabe

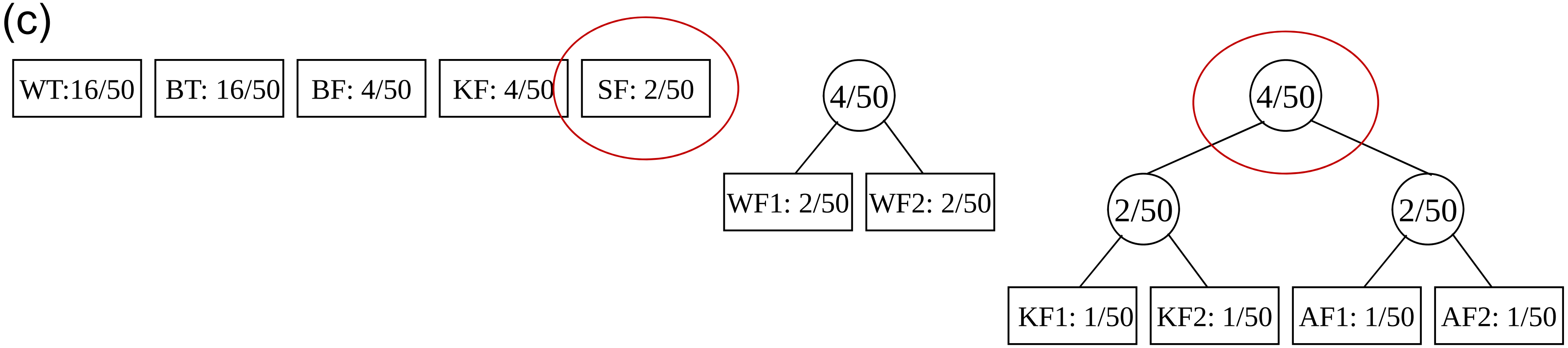
Huffman Kodierung:

(a)

(b)



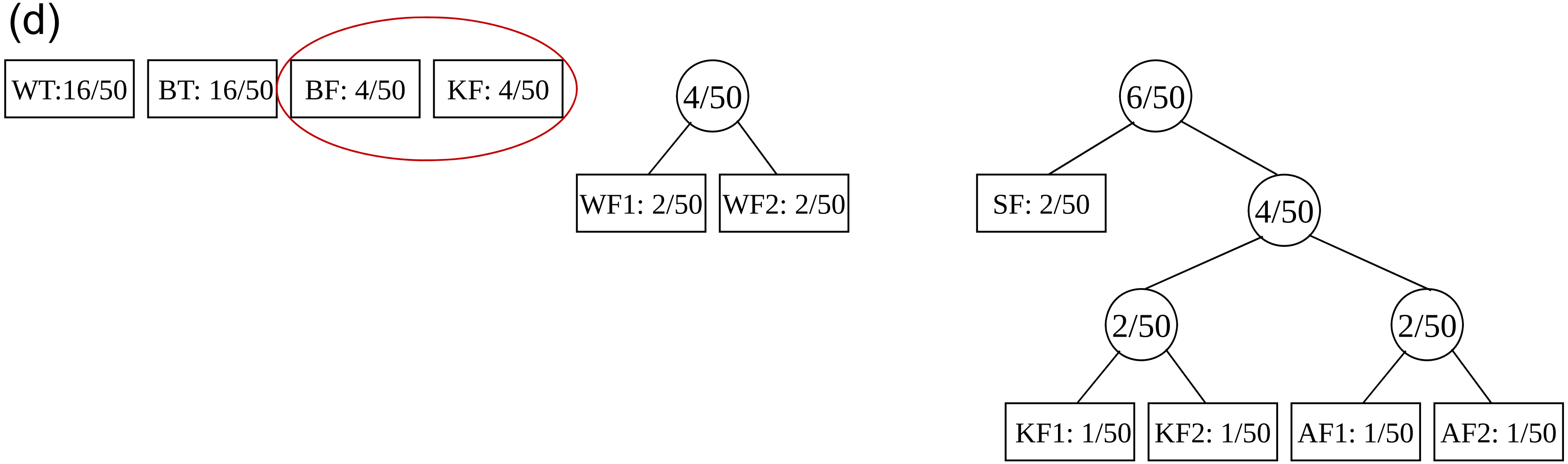
(c)



4. Aufgabe

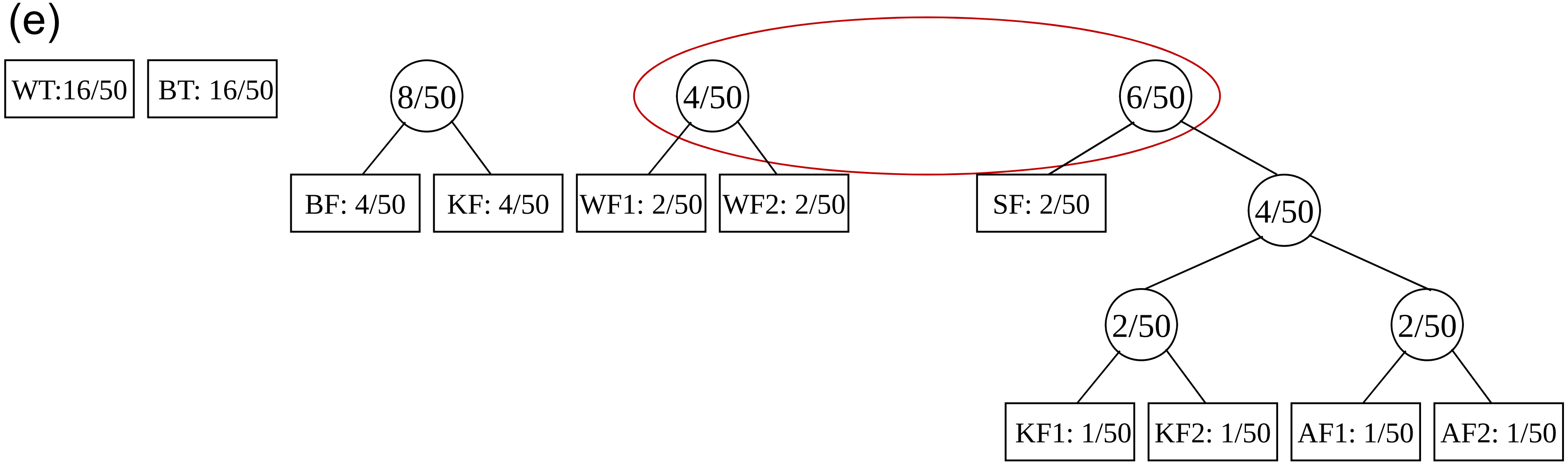
Huffman Kodierung:

(d)



Huffman Kodierung:

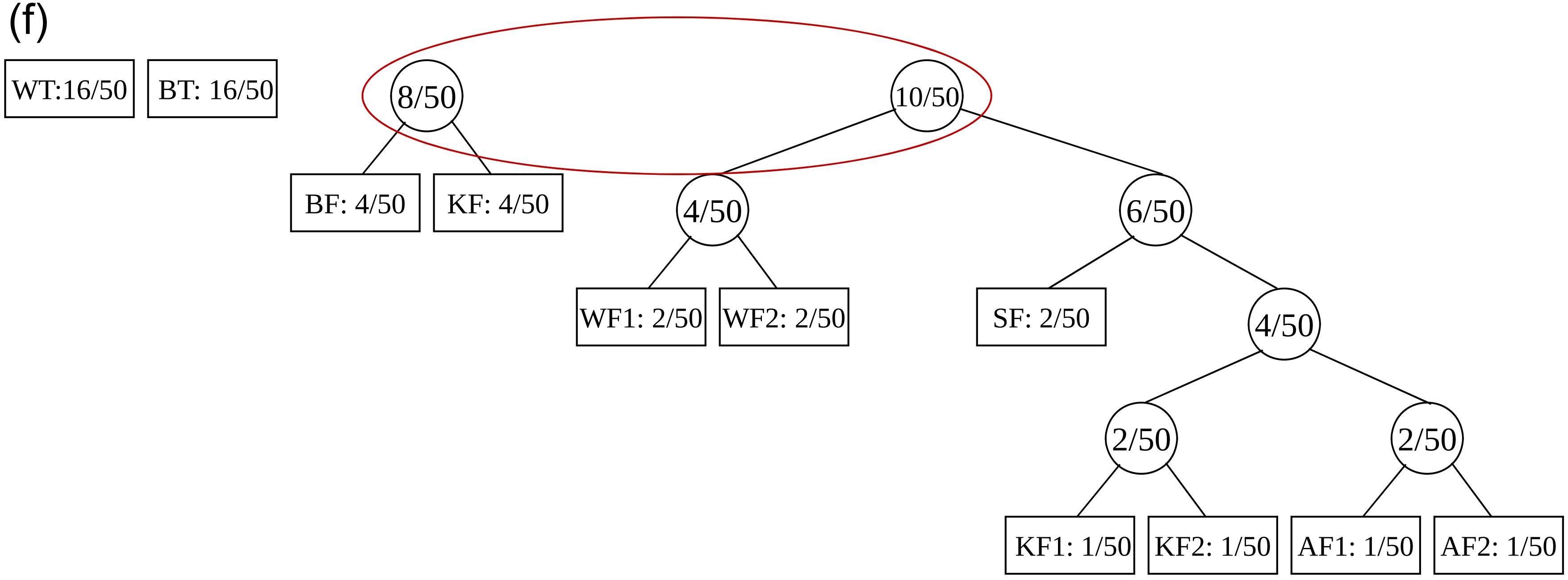
(e)



4. Aufgabe

Huffman Kodierung:

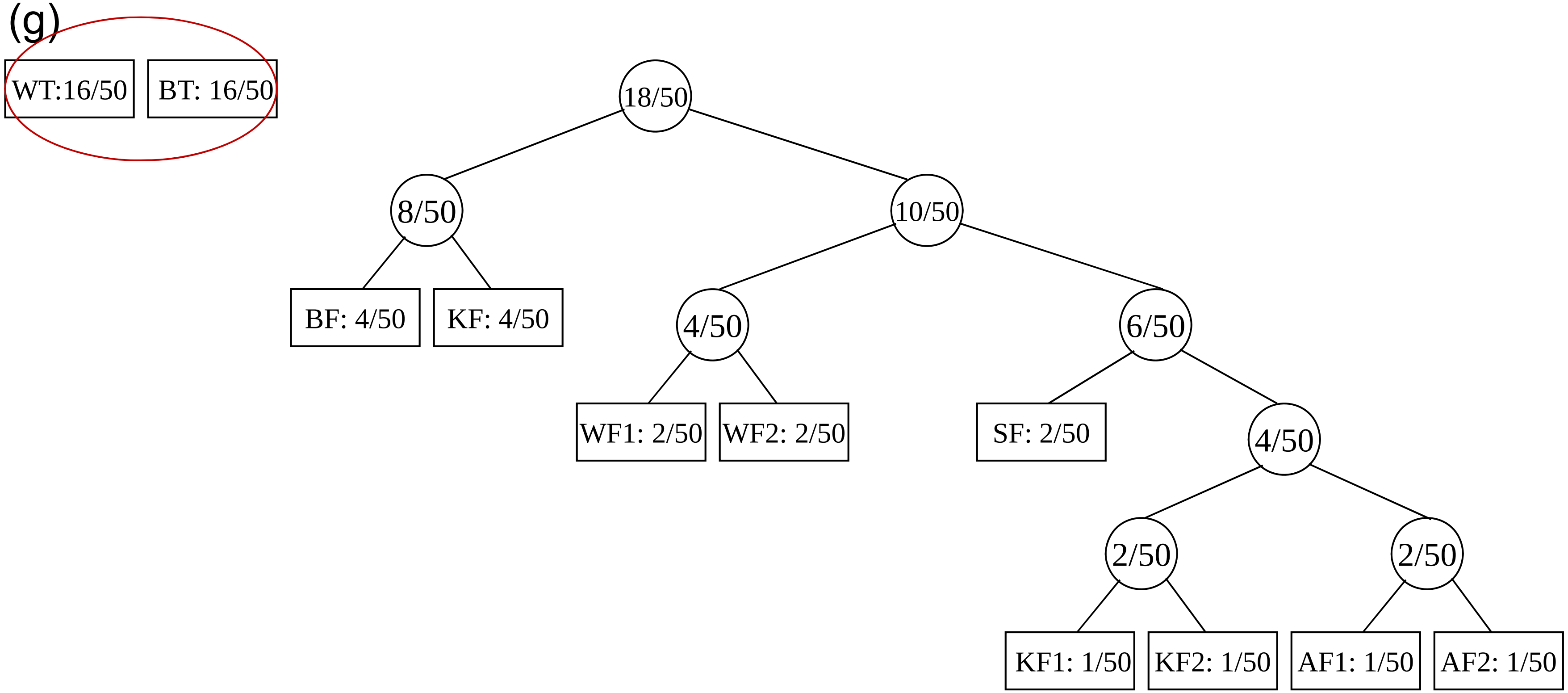
(f)



4. Aufgabe

Huffman Kodierung:

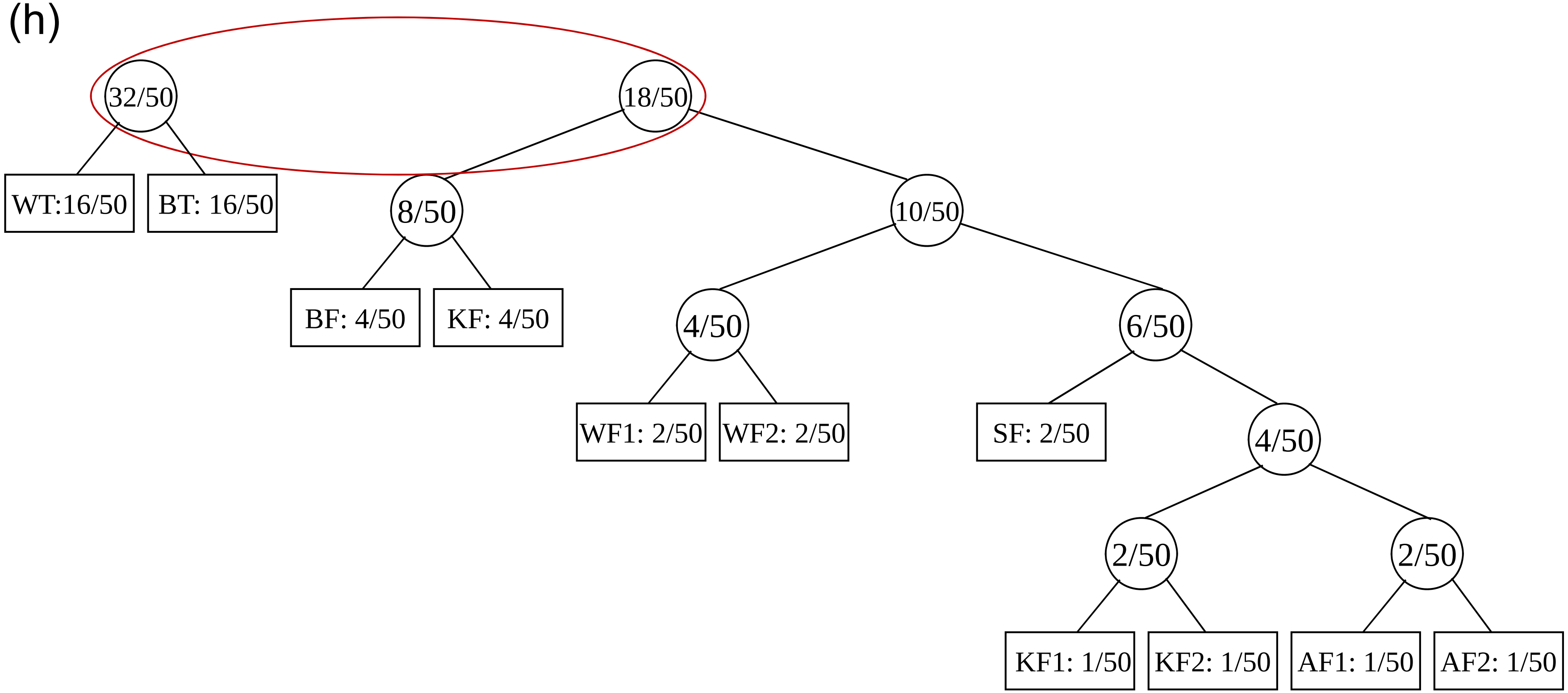
(g)



4. Aufgabe

Huffman Kodierung:

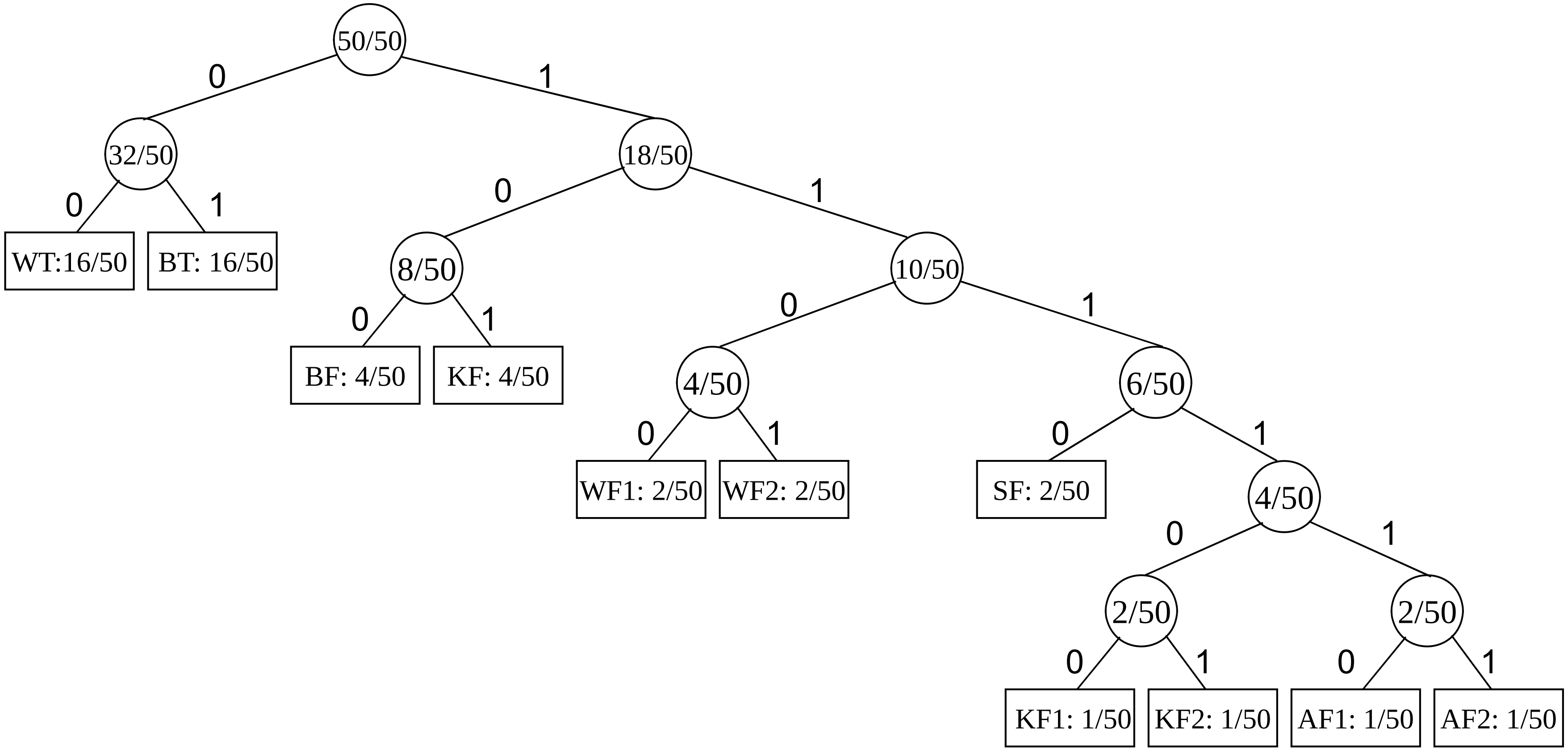
(h)



4. Aufgabe

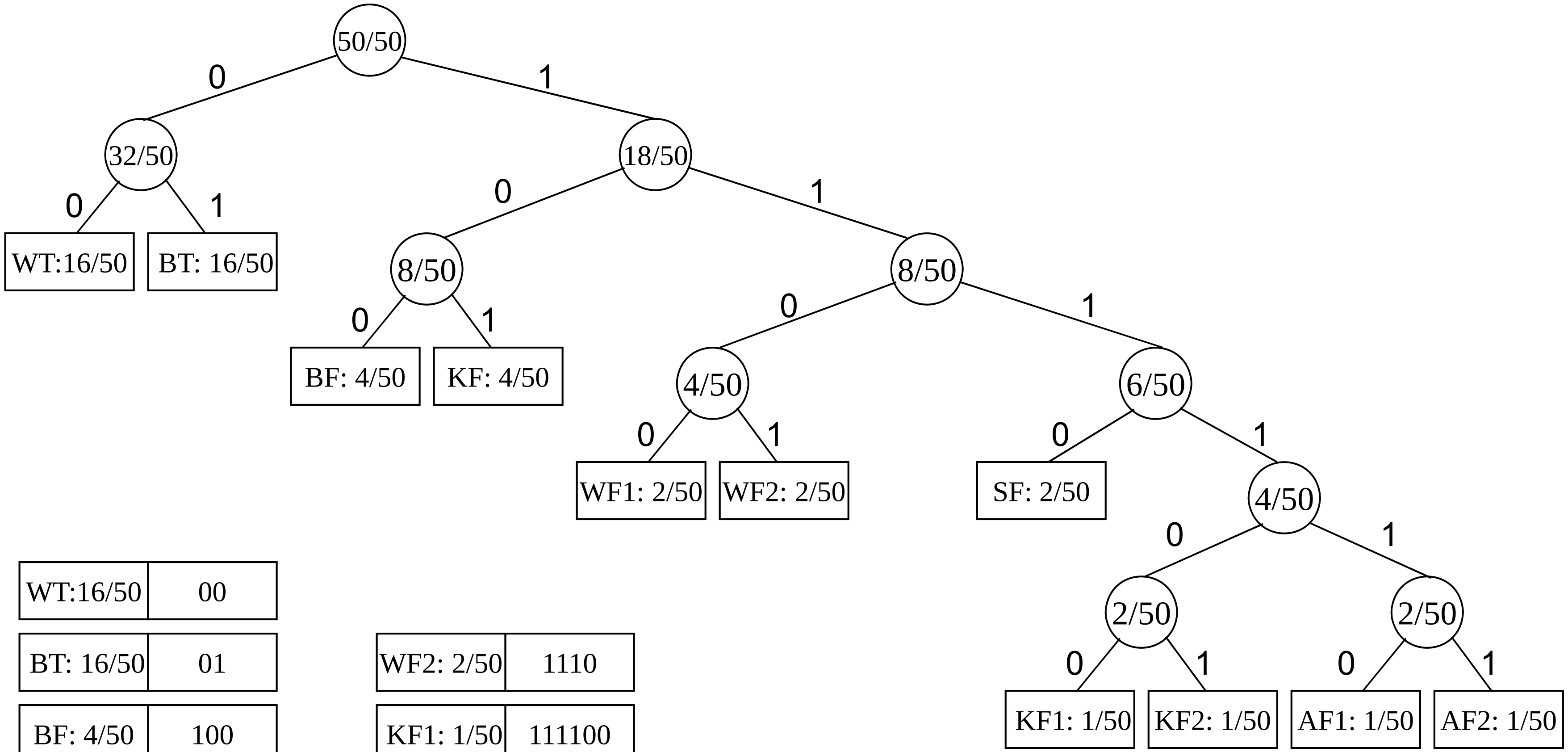
Huffman Kodierung:

(i)



4. Aufgabe

Ablesen der CWs:



WT:16/50	00
BT: 16/50	01
BF: 4/50	100
KF: 4/50	101
SF: 2/50	1100
WF1: 2/50	1101

WF2: 2/50	1110
KF1: 1/50	111100
KF2: 1/50	111101
AF1: 1/50	111110
AF2: 1/50	111111

4. Aufgabe

Ort	Häufigkeit	Wahrscheinlichkeit	Ermittelte Codierung
Wohnungstür	16	$16/50$	00
Balkontür	16	$16/50$	01
Badfenster	4	$4/50$	100
Küchenfenster	4	$4/50$	101
Schlazimmerfenster	2	$2/50$	1100
Wohnzimmerfenster 1	2	$2/50$	1101
Wohnzimmerfenster 2	2	$2/50$	1110
Kinderzimmerfenster 1	1	$1/50$	111100
Kinderzimmerfenster 2	1	$1/50$	111101
Atelierfenster 1	1	$1/50$	111110
Atelierfenster 2	1	$1/50$	111111

4. Aufgabe

4.3 Wie viele Codeworte können Sie in einem Speicher mit einer Kapazität von 1024 Bit abspeichern? Berechnen Sie dazu zuerst die mittlere Codewortlänge.

Ort	Häufigkeit	Wahrscheinlichkeit	Ermittelte Codierung
Wohnungstür	16	$16/50$	00
Balkontür	16	$16/50$	01
Badfenster	4	$4/50$	100
Küchenfenster	4	$4/50$	101
Schlazimmerfenster	2	$2/50$	1100
Wohnzimmerfenster 1	2	$2/50$	1101
Wohnzimmerfenster 2	2	$2/50$	1110
Kinderzimmerfenster 1	1	$1/50$	111100
Kinderzimmerfenster 2	1	$1/50$	111101
Atelierfenster 1	1	$1/50$	111110
Atelierfenster 2	1	$1/50$	111111

4. Aufgabe

Ort	Häufigkeit	Wahrscheinlichkeit	Ermittelte Codierung
Wohnungstür	16	$16/50$	00
Balkontür	16	$16/50$	01
Badfenster	4	$4/50$	100
Küchenfenster	4	$4/50$	101
Schlazimmerfenster	2	$2/50$	1100
Wohnzimmerfenster 1	2	$2/50$	1101
Wohnzimmerfenster 2	2	$2/50$	1110
Kinderzimmerfenster 1	1	$1/50$	111100
Kinderzimmerfenster 2	1	$1/50$	111101
Atelierfenster 1	1	$1/50$	111110
Atelierfenster 2	1	$1/50$	111111

Mittlere Codewortlänge: $$\overline{m} = \sum_{i=1}^{n} m(x_i) \cdot p(x_i)$$

$$\overline{m} = 2\left(2 \cdot 16/50\right) + 3\left(2 \cdot 4/50\right) + 4\left(3 \cdot 2/50\right) + 6\left(4 \cdot 1/50\right) = \underline{\underline{136/50 \;\; (=2{,}72)}}$$

4. Aufgabe

Ort	Häufigkeit	Wahrscheinlichkeit	Ermittelte Codierung
Wohnungstür	16	$16/50$	00
Balkontür	16	$16/50$	01
Badfenster	4	$4/50$	100
Küchenfenster	4	$4/50$	101
Schlazimmerfenster	2	$2/50$	1100
Wohnzimmerfenster 1	2	$2/50$	1101
Wohnzimmerfenster 2	2	$2/50$	1110
Kinderzimmerfenster 1	1	$1/50$	111100
Kinderzimmerfenster 2	1	$1/50$	111101
Atelierfenster 1	1	$1/50$	111110
Atelierfenster 2	1	$1/50$	111111

Anzahl speicherbarer Codeworte: $n = \dfrac{Speicherkapazität}{\overline{m}}$

$$n = \frac{1024}{136/50} = \underline{\underline{\frac{51200}{136} (\approx 376{,}5)}} \text{ CWs}$$