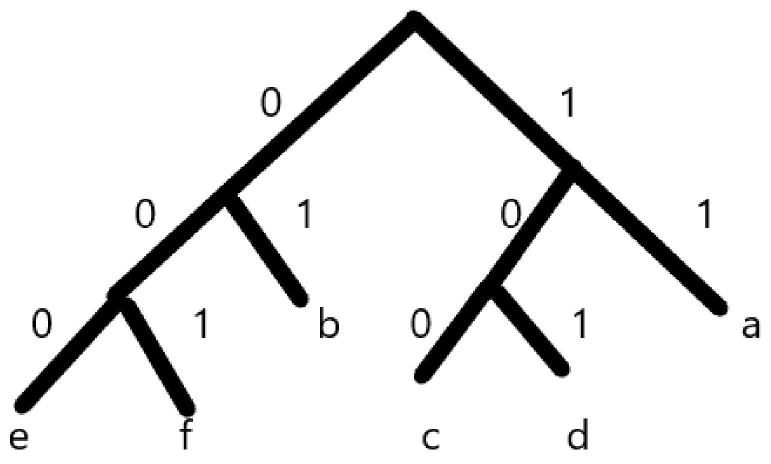
Aufgabe 5.1
a)

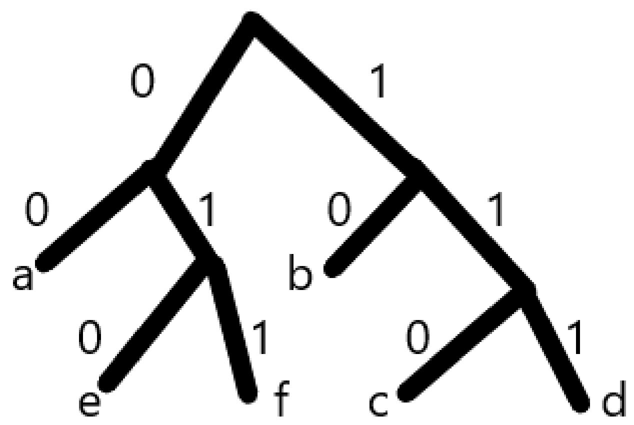


a	b	c	d	e	f
11	01	100	101	000	001

b)
Mittlere Codewortlänge:
2 * 0,3 + 2 * 0,2 + 3 * 0,15 + 3* 0,15 + 3 * 0,1 + 3 * 0,1 =
2,5

c)
badecfa
01 11 101 000 100 001 11

d)



a	b	c	d	e	f
00	10	110	111	010	011

Mittlere Codewortlänge:
2 * 0,3 + 2 * 0,2 + 3 * 0,15 + 3 * 0,15 + 3 * 0,1 + 3 * 0,1 = 2,5
badecfa:
10 00 111 010 110 011 00

Aufgabe 5.2
a) **Code-Redundanz** geht über die reine Darstellung von Codewörtern hinaus und kann zum Beispiel dazu genutzt werden um Fehler zu erkennen, oder sogar zu beheben, um dem Platzverbrauch einen Mehrwert zu schaffen.

Die Hamming-Distanz gibt an, wie viele Stellen eines Codeworts sich unterscheiden. Die niedrigste Hamming-Distanz wird als **Hamming-Abstand** bezeichnet und ist relevant für die Fehlererkennung, da von diesem die Größe der erkennbaren Fehler abhängt.

b)
Es können Fehler mit maximal d-1 Bits für den Hamming-Abstand d erkannt werden: d=4 für 3 Bit und d=2 für 1 Bit

c)
Für die Korrektur von n Bits ist ein Abstand von 2n+1 notwendig:
d=7 für 3 Bit und d=3 für 1 Bit

Aufgabe 5.3
a)
Der Fehler wird über die Berechnung aller Paritätsbits modulo 2 gebildet. Da 2-Bit-Fehler eine gerade Anzahl haben und alle geraden Zahlen modulo 2 Null ergeben, wird ein 2-Bit-Fehler nicht erkannt.

b)
0010010: 0
1111111: 1
1010101: 0
0001000: 1

c)
00100101: 11100101 (2n)+1-Bit-Fehler
11111111: 11110011 (2n)-Bit-Fehler

d)
00100101: 00100001 (1-Bit-Fehler)
11111111: 11111111 (ohne Fehler)

Aufgabe 5.4
a)

3	5	2	8	0	5	7	8 3	3 8	6
x10	x9	x8	x7	x6	x5	x4	x3	x2	x1
30	45	16	56	0	25	28	24 9	6 16	6

3-528-05783-6 ist ungültig: 236 mod 11 = 5
3-528-05738-6 ist gültig: 231 mod 11 = 0

b)

2	8	1	2	3	4	5	5	4	3	2	1	x
1	3	1	3	1	3	1	3	1	3	1	3	1
2	4	1	6	3	2	5	5	4	9	2	3	?

= 46
46 mod 10 = 4
deshalb ist x = 4

Aufgabe 5.5
a)
0|1000111
0|1100101
1|1101100 -> 1101000
0|1100101
0|1101001
1|1101101
0|0101011

b)

1000111 1100101 1101000 1100101 1101001 1101101

G e h e i m