

Lauf längencodierung

Musterlösung (Text)

- a) Code: M(1) I(1) S(2) I(1) S(2) I(1) P(2) I(1) bzw. M1I1S2I1S2I1P2I1
- b) Klartext: PPPPPPABBBBCCD.
Kompressionsrate: $\frac{15-10}{15} = 0, \bar{3} = 33, \bar{3}\%$
- c) Peters Vorschlag funktioniert leider nicht, auch wenn es verlockend erscheint, auf diese Art noch mehr Platz zu sparen. Aus der Angabe A(9) kann man nicht schlussfolgern, wie die A verteilt sind.
- d) Da immer abwechselnd Zeichen und Anzahlen gespeichert werden, ist es kein Problem, wenn ein Zeichen eine Ziffer ist. Die Ziffer kann aufgrund ihrer Position im Code nicht mit einer Anzahl verwechselt werden.
- e) siehe Computerlösung
- f) siehe Computerlösung
- g) Um 50% komprimierbar: ABBBBBBB $\rightarrow$ A(1)B(7). Kompression: $\frac{8-4}{8} = 50\%$
Um -75% komprimierbar: Text wird länger, z.B. von 8 Zeichen zu 14 Zeichen.
ABCDEF GG $\rightarrow$ A(1)B(1)C(1)D(1)E(1)F(1)G(2). Kompression: $\frac{8-14}{8} = -75\%$
Bedingung für positive Kompressionsrate: jedes Zeichen wiederholt sich im Durchschnitt mehr als 2-mal.
Mögliche Kompressionsraten: $-100\% \leq \text{Kompressionsrate} < 100\%$
(Text ohne Wiederholungen ... beliebig langer Text aus einem selben Zeichen)
- h) siehe Computerlösung
- i) Die Anzahl wird in einem Byte gespeichert. In ein Byte passt maximal die Zahl 255.
Würde man eine größere Anzahl speichern, so bräuchte man ein weiteres Byte. Das kann man aber nicht erkennbar machen, somit würde das Decodieren fehlschlagen.
- Möglichkeit 1: für jede Anzahl 2 Bytes verwenden. Dann würde man pro sich wiederholendes Zeichen 1+2 Bytes statt 1+1 benötigen
Vorteil: gut bei sehr langen Wiederholungen
Nachteil: um fast 50% längerer Code
- Möglichkeit 2: Größere Anzahlen aufspalten, z.B. statt A(300) speichert man A(255)A(45)
Vorteil: Code nicht unwesentlich länger, beliebige Wiederholungen möglich
Nachteil: Code wird vergleichsweise lang, wenn sich ein Zeichen sehr oft wiederholt

Klassenstufen 7/8

2. Aufgabe

Musterlösung (Programm)

P25L12.xlsx

P25L12eh.sb, P25L12fh.sb

Punktverteilung

a)	Code ist angegeben	1
	Lösung ist fehlerfrei	1
b)	Code	1
	Kompressionsrate	1
c)	Erkenntnis: Vorschlag ist nicht realisierbar	1
	Begründung	1
d)	Begründung	2
e)	Code wird aufgespalten	1
	Buchstaben werden wiederholt	2
	Korrektes Ergebnis. Probe: M1I1S2I1S4 → MISSISSSS	1
f)	Zerlegung in einzelne Buchstabenfolgen	3
	Codewort zusammensetzen	1
	Korrektes Ergebnis. Probe: MISSISSSS → M(1) I(1) S(2) I(1) S(4) bzw. M1I1S2I1S4	1
g)	Beispiel für 50% Kompression	1
	Beispiel für -75% Kompression	2
	Bedingung für positive Kompression	1
	Untere Grenze: $-100\% \leq \text{Kompressionsrate}$	1
	Obere Grenze: $\text{Kompressionsrate} < 100\%$	1
h)	Ermitteln der Längen von Klartext und Code	1
	Ausgabe der Kompressionsrate. Probe: MISSISSSS → Kompression: -0,11	1
	Ausgabe einer Warnung bei negativer Kompressionsrate	1
i)	In ein Byte passt maximal die Zahl 255	1
	sinnvoller Lösungsvorschlag	1
	Vorteil	1
	Nachteil	1
Gesamt		30