Grundlagen der Informatik

Prof. Dr. J. Schmidt
Fakultät für Informatik

GDI – WS 2018/19

Information und Quellencodierung
Lauflängencodierung / LZW-Kompression

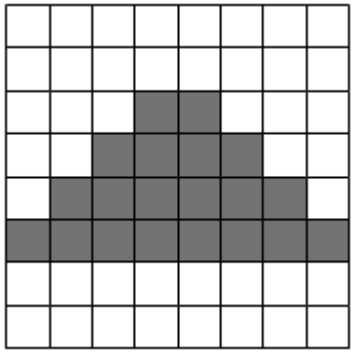
Hochschule **Rosenheim**
University of Applied Sciences

Lauflängen-Kodierung (1)

- „RLE“ / „RLC“
  - Engl. „run-length-encoding“ / „run-length-coding“
- Eindimensionale Lauflängen-Kodierung
  - Ergänzung eines Zählers
  - zusätzlich zu den kodierten Daten
- Speicherung
  - Lauflänge = Anzahl der Wiederholungen eines Zeichens
  - Speicherung von Zahlenpaaren etwa der Art (f,n)
    - f gibt den Datenwert an und
    - n gibt die Lauflänge an
      (gerechnet ab Anfang des Datenstroms bzw. ab Ende der vorherigen Sequenz)

Lauflängen-Kodierung (2)

- Einfaches Beispiel
  - Lauflängen-Kodierung eines Binärbildes



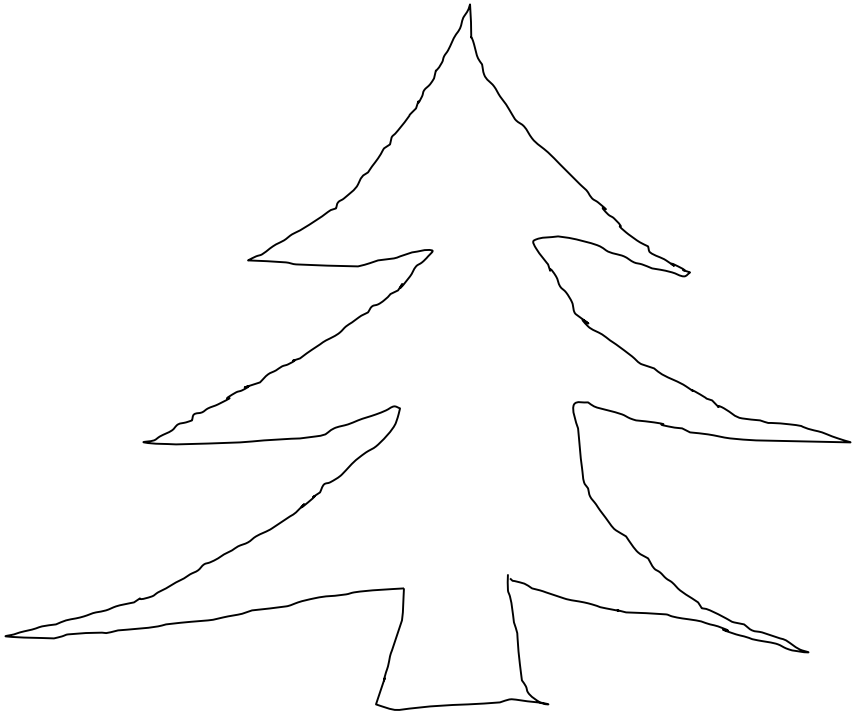
```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 0 0 0
0 0 1 1 1 1 0 0
0 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

```
0 0 0 0
0 0 0 0
0 0 1 1 1 0 1 0 0 0 1 1
0 0 1 0 1 1 0 0 0 0 1 0
0 0 0 1 1 1 1 0 0 0 0 1
1 0 0 0
0 0 0 0
0 0 0 0
```

  - Übertragung Zahlenpaare (Datenwert, Lauflänge)
    - Lauflängen: 001=1 010=2 011=3 100=4
      101=5 110=6 111=7 000=8
  - Kompression von 64 auf 56 Bit wird erreicht

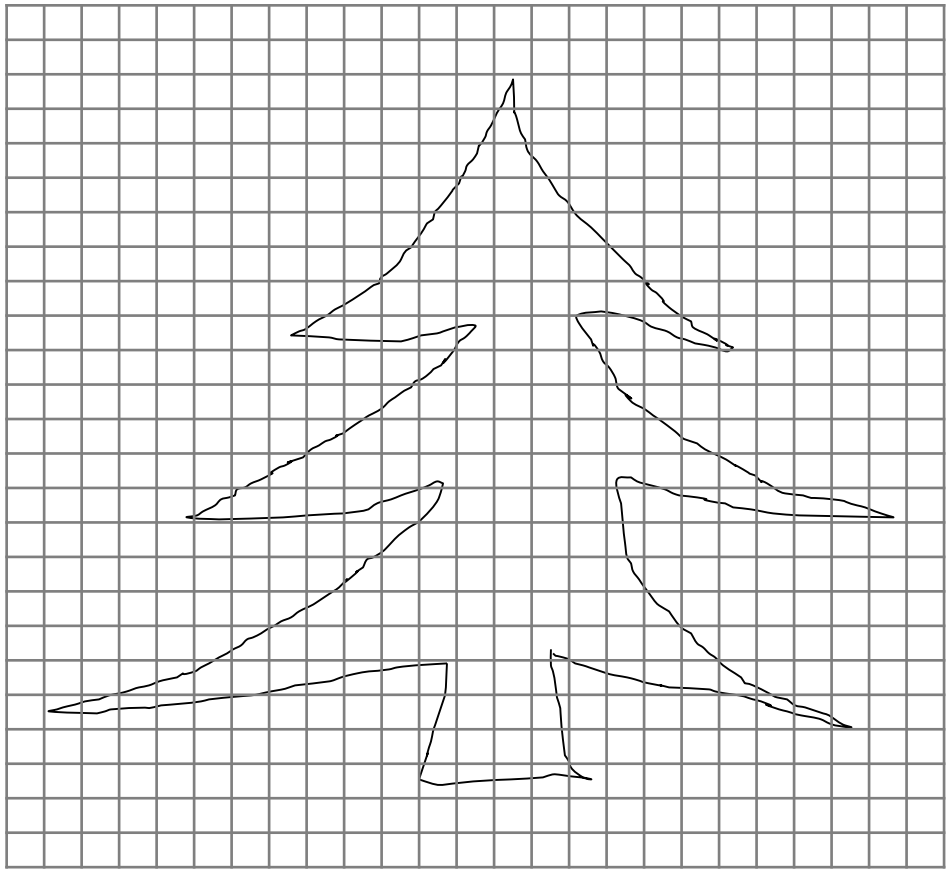
Lauflängen-Kodierung (3)

- Anwendungsbeispiel: Fax schicken
  - Original



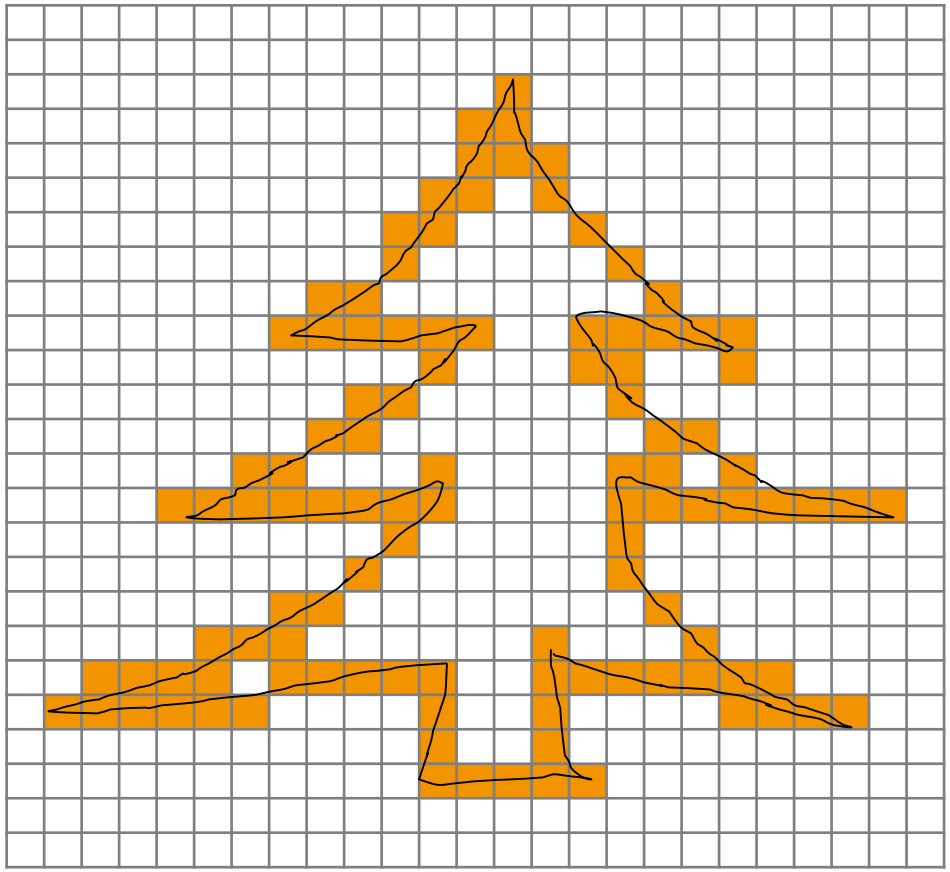
Lauflängen-Kodierung (4)

- Anwendungsbeispiel: Fax schicken
  - Gitter-netz



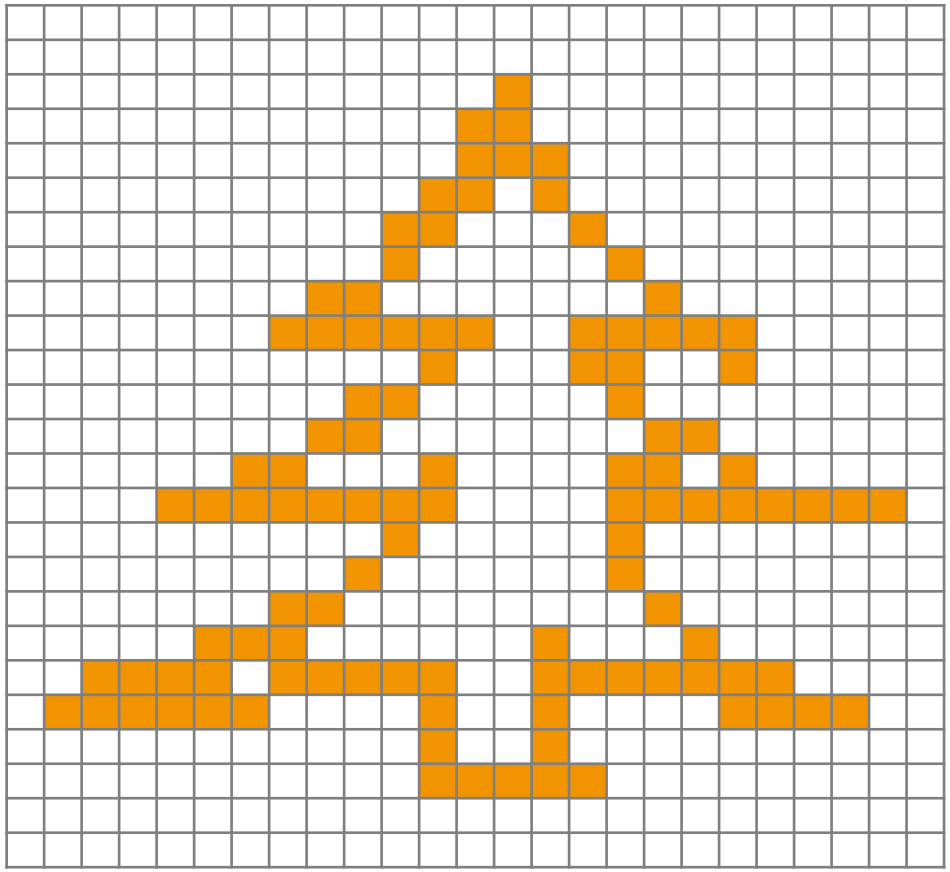
Lauflängen-Kodierung (5)

- Anwendungsbeispiel: Fax schicken
  - Raster-bild



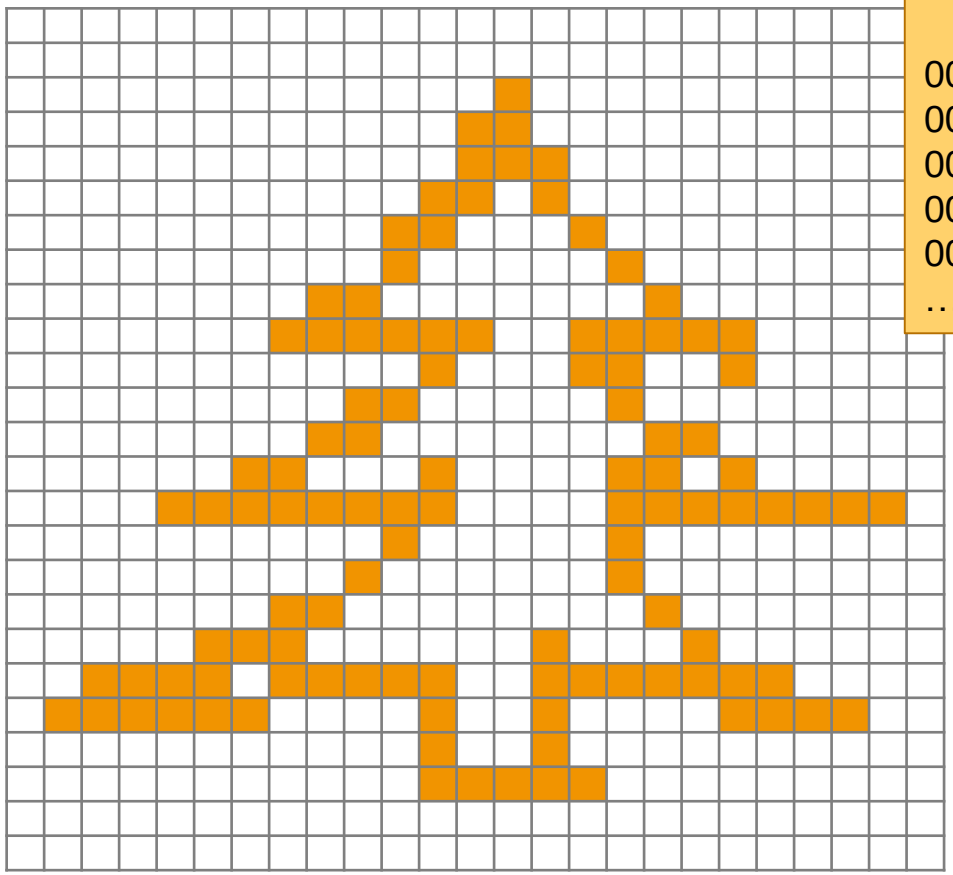
Lauflängen-Kodierung (6)

- Anwendungsbeispiel: Fax schicken
  - Raster-punkt = Spei-cher-stelle



Lauflängen-Kodierung (7)

- Anwendungsbeispiel: Fax schicken
  - Speicherung
  - 625 Bit

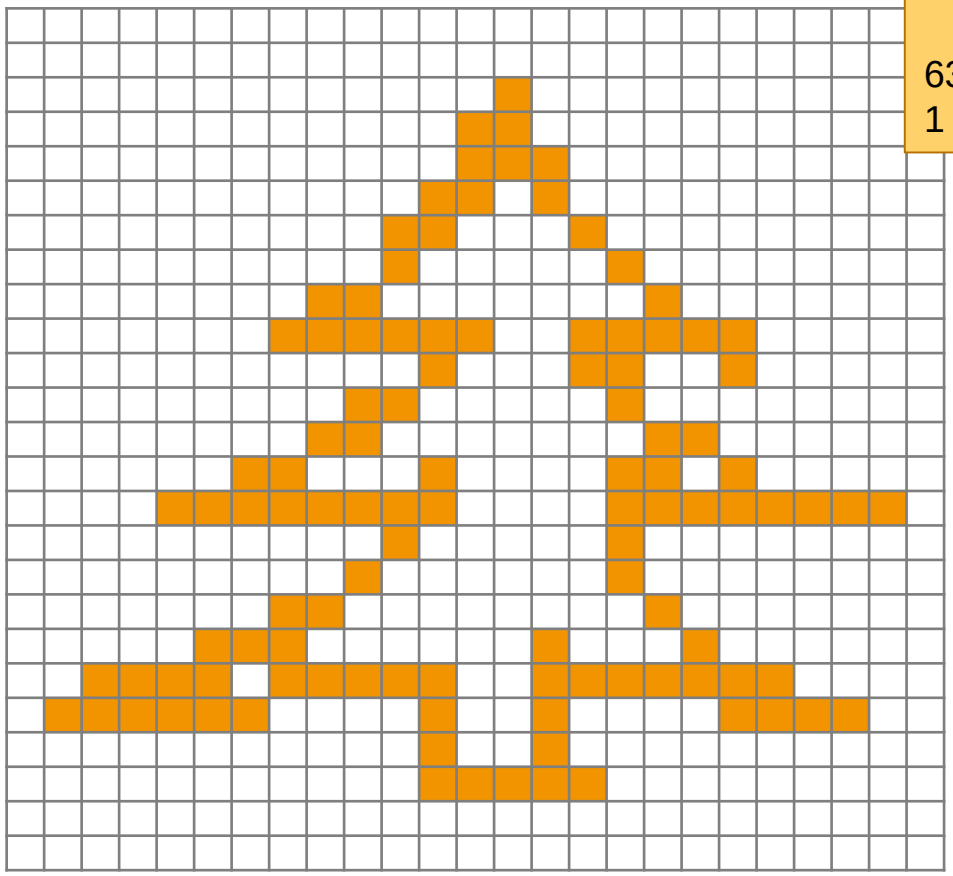


0 = weiß
1 = schwarz

0000000000000000000000000
0000000000000000000000000
0000000000000100000000000
0000000000001100000000000
0000000000011110000000000
…

Lauflängen-Kodierung (8)

- Anwendungsbeispiel: Fax schicken
  - Lauf-längen



0 = weiß
1 = schwarz

63 1 23 2 23 3 21 2 1 1 20 2 3
1 19 1 5 1 16 2 7 1 …

Lauflängen-Kodierung (9)

- Anwendungsbeispiel: Fax schicken
  - Lauflängen
    63 1 23 2 23 3 21 2 1 1 20 2 3 1 19 1 5 1 16 2 7 1 …
  - Weitere Verbesserung
    - Berücksichtigung der Häufigkeiten der Lauflängen
    - Code-Baum mit variabler Code-Länge
  - Faxgeräte
    - Eingespeicherte „Standard Code-Bäume“

Lauflängen-Kodierung (10)

- Beobachtungen
  - Effiziente Komprimierung nur möglich,
    - falls in den Daten zahlreiche homogene Bereiche auftreten,
    - die durch ein einziges Code-Wort charakterisiert werden können
  - Anwendung v.a. in
    - computergenerierten Bildern und Grafiken
    - sowie Binärbilder mit zwei Helligkeitsstufen
    - oft in Kombination mit Huffman (z.B. in JPEG)
  - Vergrößerung der Datei möglich
    - Bei wenigen längeren Sequenzen von identischen Daten

LZW-Algorithmus (1)

- Erfinder des Algorithmus
  - **L**empel, **Z**iv und **W**elch (1978/83)
- Zugrunde liegende Idee
  - Erweiterung RLC:
    Nicht nur Kodierung von Einzelzeichen, sondern auch von Zeichengruppen unterschiedlicher Länge
  - Berücksichtigung von Häufigkeiten der Einzelzeichen sowie Redundanzen, die auf der Korrelation aufeinander folgender Zeichen beruhen
  - Erkennen von redundanten Zeichenfolgen, die durch kürzeren Code ersetzt werden

LZW-Algorithmus (2)

- Auswirkungen
  - LZW-Algorithmus minimiert die Redundanzen, die dadurch entstehen, dass identische Zeichenfolgen sich in den Eingabedaten mehrmals wiederholen
  - Die Kompressionswirkung ist umso besser, je häufiger solche Wiederholungen auftreten und je länger die sich wiederholenden Zeichenfolgen sind
  - Ergebnis ist eine weitgehend unkorrelierte Zeichenfolge, die verlustfrei nicht mehr weiter komprimierbar ist

LZW-Algorithmus (3)

- Prinzip
  - Dynamische Generierung einer Code-Tabelle
  - Jeder Eintrag besteht aus
    - einer Zeichenfolge mit Zeichen aus dem Quell-Alphabet
    - und dem zugehörigen komprimierten Code
  - Code-Tabelle
    - wird zu Beginn mit allen Einzelzeichen des Quell-Alphabets vorbesetzt
    - wird während der Kompression nach und nach erweitert und an die Eingabe angepasst

LZW-Algorithmus (4)

- Charakteristische Eigenschaften
  - Benötigt keine Informationen über die Statistik des Eingabetextes
    - (z.B. Auftrittswahrscheinlichkeiten der Einzelzeichen)
  - Code-Tabelle muss nicht zusammen mit den kodierten Daten gespeichert bzw. übertragen werden
    - Wird bei Dekodierung aus den kodierten Daten in identischer Weise wieder erzeugt

LZW-Algorithmus (5)

- Anwendung
  - Bild- und Dokumentformate
    - Zum Sparen von Speicherplatz
  - Beispiele
    - GIF, TIFF, PDF, Postscript

LZW-Algorithmus (6)

- LZW-Kompression eines Strings Z
  - Initialisiere die Code-Tabelle mit den Einzelzeichen
  - Weise dem Präfix P den Leerstring zu
  - Wiederhole, solange Eingabezeichen vorhanden sind:
    - Lies nächstes Eingabezeichen c aus dem Eingabestring Z
    - Wenn Pc in der Code-Tabelle gefunden wird:
      - Setze $P=Pc$
    - Sonst:
      - Trage Pc in die nächste freie Position der Code-Tabelle ein
      - Gib den Code für P aus
      - Setze $P=c$
  - Ende der Schleife
  - Gib den Code für das letzte Präfix P aus

LZW-Algorithmus (7)

- Beispiel: Kodierung der Zeichenkette Z = ABABCBABAB
  - Vorbereitung
    - Initialisiere die Code-Tabelle mit den Einzelzeichen

Präfix	Code
A	0
B	1
C	2

LZW-Algorithmus (8)

- Beispiel: Kodierung der Zeichenkette Z = ABABCBABAB

  - Kodierungsvorgang

Zeichen c	Präfix P	Ausgabe
	-	
A	A	
B	B	0
A	A	1
B	AB	
C	C	3
B	B	2
A	BA	
B	B	4
A	BA	
B	BAB	
		7

Code-Tabelle

Präfix	Code
A	0
B	1
C	2
AB	3
BA	4
ABC	5
CB	6
BAB	7

Lies nächstes Eingabezeichen c
Wenn (Pc in der Code-Tabelle)
  Setze $P=Pc$
Sonst
  Trage Pc in Code-Tabelle ein
  Gib den Code für P aus
  Setze $P=c$

Gib den Code für
das letzte Präfix P aus

➔ Kodierte Nachricht: **013247**

LZW-Algorithmus (9)

- LZW-Dekompression einer Nachricht
  - Initialisiere die Code-Tabelle mit den Eingabezeichen
  - Weise dem Präfix P den Leerstring zu
  - Wiederhole, solange Eingabezeichen vorhanden sind:
    - Lies nächstes Eingabezeichen c
    - Wenn c in der Code-Tabelle enthalten ist:
      - Gib den zu c gehörenden String aus
      - Setze k = erstes Zeichen dieses Strings
      - Trage Pk in die Code-Tabelle ein, falls noch nicht enthalten
      - Setze P = zum Code c gehörender String
    - Sonst (Sonderfall):
      - Setze k = erstes Zeichen von P
      - Gib Pk aus
      - Trage Pk in die Code-Tabelle ein
      - Setze $P = Pk$
  - Ende der Schleife

LZW-Algorithmus (10)

- Beispiel: Dekomprimierung Kompressions-Ergebnis 013247
  - Dekodierungsvorgang

Code c	k	Präfix P	Ausgabe
		-	
0	A	A	A
1	B	B	B
3	A	AB	AB
2	C	C	C
4	B	BA	BA
7	B	BAB	BAB

Code-Tabelle

Präfix	Code
A	0
B	1
C	2
AB	3
BA	4
ABC	5
CB	6
BAB	7

Lies nächstes Eingabezeichen c
Wenn (c in Code-Tabelle)
  Gib den zu c gehörenden String aus
  Setze k = erstes Zeichen dieses Strings
  Trage Pk in Code-Tabelle ein, falls nicht enth.
  Setze P = zum Code c gehörender String
Sonst
  Setze k = erstes Zeichen von P
  Gib Pk aus
  Trage Pk in die Code-Tabelle ein
  Setze $P = Pk$

➔ Dekodierte Nachricht: ABABCBABAB

Aufgabe

- Kodieren Sie die Zeichenkette ABBABABAC mit Hilfe des LZW-Algorithmus.

Lies nächstes Eingabezeichen c
Wenn (Pc in der Code-Tabelle)
  Setze P=Pc
Sonst
  Trage Pc in Code-Tabelle ein
  Gib den Code für P aus
  Setze P=c

Gib den Code für
das letzte Präfix P aus