PROFESSOR ERKLÄRBÄR

Serie Informatik · Folge 3 · Advanced

INFORMATIK

LEHRERINFO ZUM VIDEO

Codierung, Kompression und Fehlerkorrektur

FACH	STUFE	DAUER	VIDEO
Informatik	Gymnasium · Sek II	14 min	youtube.com/@Erklaerbaer

01 Lernziele

Nach dem Video können die Schülerinnen und Schüler …

- Information als Antwort auf Ja-Nein-Fragen deuten und den Begriff Entropie als Grenze der verlustfreien Kompression erklären,
- ein Zeichen nach UTF-8 codieren und den Aufbau aus Start- und Folgebytes begründen,
- einen Huffman-Baum aufstellen, die Codes ablesen und die Ersparnis gegenüber ASCII berechnen,
- verlustfreie und verlustbehaftete Kompression unterscheiden und die Prinzipien von JPEG und MP3 beschreiben,
- Paritätsbit und Prüfziffern (EAN, IBAN) zur Fehlererkennung anwenden,
- mit dem Hamming-Code (7,4) ein Codewort bilden und einen Ein-Bit-Fehler finden und korrigieren.

02 Voraussetzungen

ASCII, Unicode, Pixel, Abtastung und Lauflängencodierung (Folge 3 · Grundlagen); Hexadezimalzahlen und Rechnen mit Bits (Folge 2 · Advanced). Zweierpotenzen und Logarithmus zur Basis 2 sind hilfreich, aber nicht nötig.

03 Ablauf des Videos

ZEIT	KAPITEL	INHALT UND BEISPIEL
0:00	Einstieg	QR-Code mit Loch funktioniert noch (bis 30 %) · Ident · drei Fragen
0:30	Was du brauchst	ASCII, Unicode, RGB, Abtastwerte, Lauflängencodierung, Hex · Hinweis auf 03G
1:01	Wie viel Information steckt drin?	Bit = Ja-Nein-Frage · Zahlenraten 1–100 in 7 Fragen · seltene Ereignisse · Entropie · faire und gezinkte Münze

2:13	UTF-8 im Detail	Startbits 0 / 110 / 1110 / 11110, Folgebytes 10 · ä = 228 → C3 A4 · Umlaut-Fehler PAUSE-FRAGE Bytes für «Bär»
3:33	Huffman-Code	ABRAKADABRA: Baum, Codes, 23 statt 88 Bits · präfixfrei · Entropie ≈ 22,4 Bits · Schubfachprinzip PAUSE-FRAGE ANANAS
5:33	Verlustbehaftet komprimieren	JPEG: 8×8-Blöcke, feine Details und Farbe gröber · Klötzchen im Messenger · MP3/AAC: Verdeckung · Video: nur Unterschiede
6:46	Fehler erkennen	Paritätsbit am «A» · EAN-Prüfziffer (Summe 110 → 0) · IBAN (Division durch 97) · erkennen ≠ korrigieren
8:01	Fehler korrigieren	Hamming (7,4): Prüfbits an 1, 2, 4 · Daten 1011 → 0110011 · Bit 5 kippt → Gruppen 1 + 4 = 5 · Reed-Solomon im QR-Code, CD, Raumsonden
9:31	Projekt-Baustein	Befehle sind Codes: 4 Bit Befehl + 4 Bit Adresse · «lade aus Zelle 14» = 0001 1110 = 1E
9:53	Prüfungsfallen	fünf Fallen (siehe 04)
10:47	Zusammenfassung	Kernaussagen · zurück zum QR-Code mit Loch
11:26	Quiz	4 Fragen · Endkarte Folge 4 Advanced

04 Prüfungsfallen

- Ein Huffman-Code darf beliebig sein, Hauptsache kurz – er muss präfixfrei sein, sonst ist das Decodieren nicht eindeutig.
- UTF-8 braucht immer zwei Bytes pro Zeichen – es sind ein bis vier Bytes; ASCII-Zeichen bleiben bei einem.
- Ein Paritätsbit findet jeden Fehler – kippen zwei Bits, bleibt die Anzahl der Einsen gerade.
- Jede Datei lässt sich verlustfrei verkleinern – zufällige oder schon komprimierte Daten nicht (Schubfachprinzip).
- Verlustbehaftet heisst schlecht – bei passender Einstellung ist der Unterschied nicht wahrnehmbar; es kommt auf den Zweck an.

05 Einsatz im Unterricht

Einstieg: Einen ausgedruckten QR-Code mit Filzstift teilweise schwärzen oder einreissen und scannen lassen – wie viel darf fehlen?

Spiel: Zahlenraten 1–100 in Partnerarbeit; anschliessend begründen, warum 7 Fragen immer reichen.

Huffman von Hand: Jede Gruppe codiert den eigenen Vornamen und vergleicht die Bitzahl mit ASCII.

Prüfungsvorbereitung: Aufgaben 3, 4 und 6 sind im Stil schriftlicher Prüfungen gestellt.

Hinweis: JPEG ist vereinfacht dargestellt («grob und fein verändern» steht für die Frequenzanteile der diskreten Kosinustransformation). Bei gleich häufigen Zeichen gibt es mehrere gleich gute Huffman-Bäume; die Gesamtlänge ist immer dieselbe. Die 30 % beim QR-Code gelten für die höchste Fehlerkorrekturstufe H und beziehen sich auf die Codewörter.

06 Quellen

- Claude E. Shannon: „A Mathematical Theory of Communication“, Bell System Technical Journal 27 (1948)
- David A. Huffman: „A Method for the Construction of Minimum-Redundancy Codes“, Proceedings of the IRE 40 (1952)
- Richard W. Hamming: „Error Detecting and Error Correcting Codes“, Bell System Technical Journal 29 (1950)
- RFC 3629: UTF-8, a transformation format of ISO 10646 (2003) · The Unicode Standard, Version 16.0
- ISO/IEC 18004: QR Code bar code symbology specification · GS1 General Specifications (Prüfziffer) · ISO 13616 / ISO 7064 (IBAN)
- Gesellschaft für Informatik: Bildungsstandards Informatik für die Sekundarstufe II (2016)

NAME _____ KLASSE _____ DATUM _____

Codierung, Kompression und Fehlerkorrektur

1 Information und Parität ●○○

a) Wie viele Ja-Nein-Fragen braucht man höchstens, um eine Zahl von 1 bis 1000 zu erraten? **Begründe**. b) **Ergänze** für B = 100 0010 und C = 100 0011 je das Paritätsbit (gerade Parität) und **erkläre**, warum zwei gekippte Bits unbemerkt bleiben.

2 UTF-8 ●●○

a) Das ö hat die Unicode-Nummer 246. **Codiere** es in UTF-8 (binär und hexadezimal). b) Wie viele Bytes braucht das Wort «Käse»? c) **Entscheide**, ob 1011 0010 ein Startbyte oder ein Folgebyte ist.

3 Huffman-Code ●●○

Erstelle für das Wort MISSISSIPPI einen Huffman-Baum, **gib** die Codes **an** und **berechne** die Bitzahl. **Vergleiche** mit ASCII und mit einem festen Code von 2 Bits pro Zeichen.

4 Prüfziffer an der Kasse ●●●

a) **Berechne** die Prüfziffer der EAN 4 0 0 6 3 8 1 3 3 3 9 3 _ (Gewichte 1, 3, 1, 3, …). b) **Zeige**, dass das Vertauschen der Ziffern 6 und 3 an Stelle 4 und 5 erkannt wird, das Vertauschen von 3 und 8 an Stelle 5 und 6 aber nicht.

5 Verlustfrei oder verlustbehaftet? ●○○

Ordne zu und **begründe**: Programmdatei, Urlaubsfoto, Podcast, Textdokument, Röntgenbild für die Ärztin, Sprachnachricht.

6 Hamming-Code (7,4) ●●●

a) **Bilde** das Codewort für die Daten 0110 (Prüfbits an Stelle 1, 2 und 4, gerade Parität). b) Empfangen wird 1100100. **Bestimme** die fehlerhafte Stelle und die korrigierten Daten.

Lösungen

1 Information und Parität

a) 10 Fragen: $2^9 = 512 < 1000 \leq 1024 = 2^{10}$; jede Frage halbiert den Bereich. b) B: zwei Einsen → Prüfbit 0 · C: drei Einsen → Prüfbit 1. Kippen zwei Bits, ändert sich die Anzahl der Einsen um 0 oder 2 – sie bleibt gerade.

2 UTF-8

a) 246 = 000 1111 0110 (11 Bit) → 110**00011** 10**110110** = C3 B6. b) K 1 + ä 2 + s 1 + e 1 = 5 Bytes. c) Es beginnt mit 10 → Folgebyte.

3 Huffman-Code

Häufigkeiten: I 4, S 4, P 2, M 1. M + P = 3 → 3 + I = 7 → 7 + S = 11. Codes z. B. S = 0, I = 10, P = 111, M = 110. Bits: 4 · 1 + 4 · 2 + 2 · 3 + 1 · 3 = 21. ASCII: 11 · 8 = 88 · fester 2-Bit-Code: 22. (Andere gleich gute Bäume sind möglich, die Summe bleibt 21.)

4 Prüfziffer an der Kasse

a) 4 + 0 + 0 + 18 + 3 + 24 + 1 + 9 + 3 + 9 + 9 + 9 = 89 → nächste Zehnerzahl 90 → Prüfziffer 1 (EAN 4006381333931). b) 6 und 3: vorher 6 · 3 + 3 · 1 = 21, nachher 3 · 3 + 6 · 1 = 15 → Summe ändert sich um 6, Fehler erkannt. 3 und 8: vorher 3 · 1 + 8 · 3 = 27, nachher 8 · 1 + 3 · 3 = 17 → Unterschied 10, die Prüfziffer bleibt gleich – nicht erkannt (Ziffern mit Abstand 5).

5 Verlustfrei oder verlustbehaftet?

Verlustfrei: Programmdatei und Textdokument (jedes Bit zählt), Röntgenbild (Details dürfen nicht verschwinden). Verlustbehaftet: Urlaubsfoto, Podcast, Sprachnachricht (Unterschied kaum wahrnehmbar, Datei viel kleiner).

6 Hamming-Code (7,4)

a) d3 = 0, d5 = 1, d6 = 1, d7 = 0 · p1 = d3 ⊕ d5 ⊕ d7 = 1 · p2 = d3 ⊕ d6 ⊕ d7 = 1 · p4 = d5 ⊕ d6 ⊕ d7 = 0 → Codewort 1100110. b) Gruppe 1 (1, 3, 5, 7): 1 + 0 + 1 + 0 = 2 ✓ · Gruppe 2 (2, 3, 6, 7): 1 + 0 + 0 + 0 = 1 ✗ · Gruppe 4 (4, 5, 6, 7): 0 + 1 + 0 + 0 = 1 ✗ → 2 + 4 = 6: Bit 6 kippen → 1100110, Daten 0110.

Q Quiz im Video

1 C (vier Bytes) · 2 A (kein Codewort ist der Anfang eines anderen) · 3 B (nur erkennen) · 4 A (Fehlerkorrektur, Reed-Solomon)