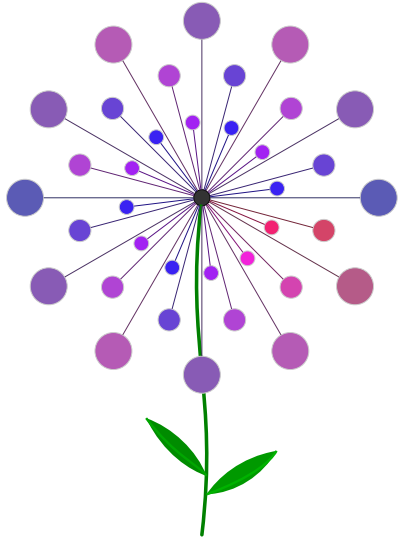


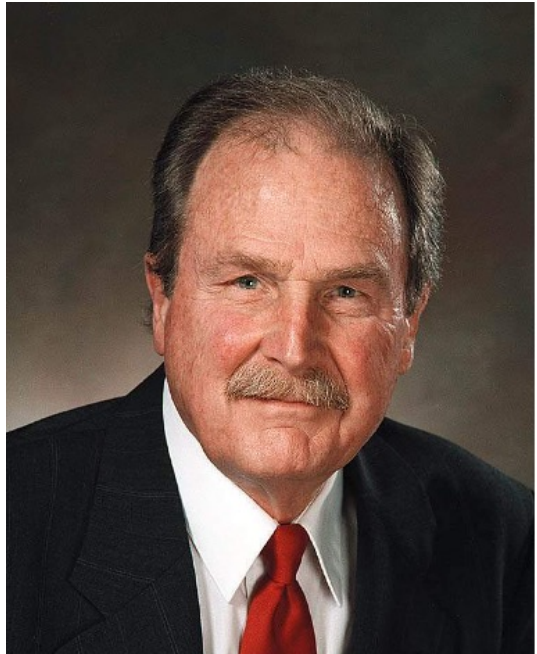
FreeFlower

Informatik: Data Science und Sicherheit

Kompression: Arithmetische Kompression

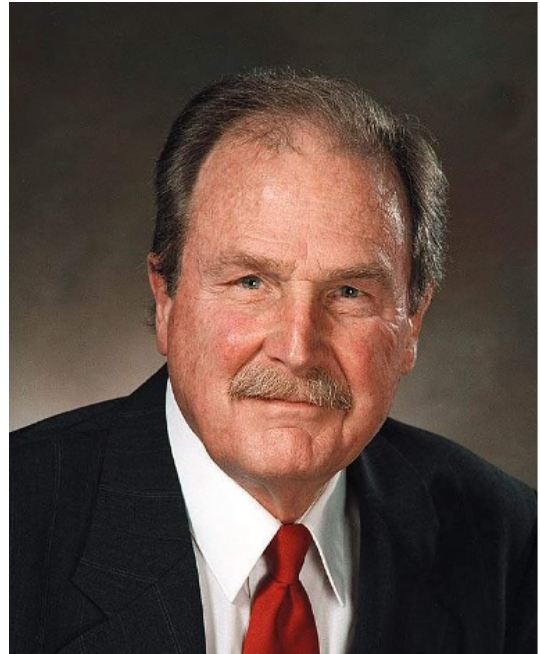
Huffman-Kodierung

- Optimale Kodierung von Zeichen.



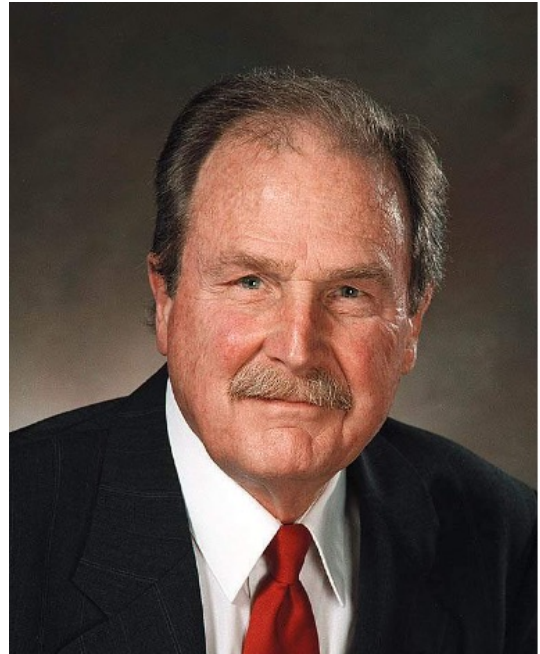
Huffman-Kodierung

- Optimale Kodierung von Zeichen.
- Wenn die Häufigkeiten der Zeichen bei 1/2, 1/4, 1/8, etc. liegen, gibt es keine bessere Kodierung



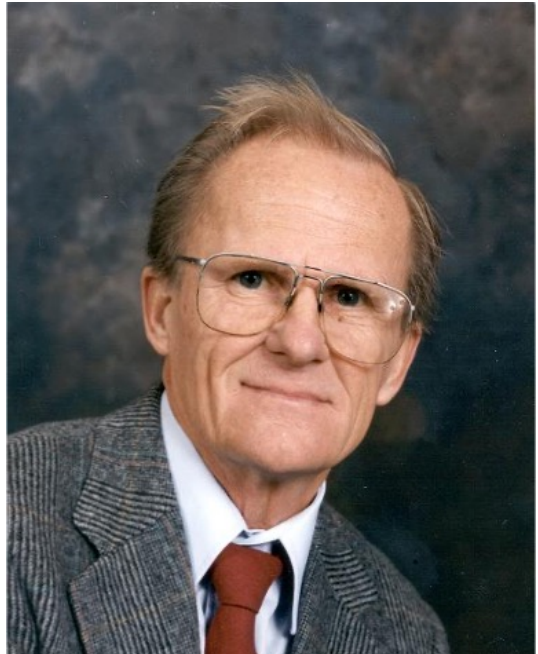
Huffman-Kodierung

- Optimale Kodierung von Zeichen.
- Wenn die Häufigkeiten der Zeichen bei 1/2, 1/4, 1/8, etc. liegen, gibt es keine bessere Kodierung
- Wenn eine Häufigkeit aber z.B. bei 30% liegt, sind 2 Bits (optimal für 25%) verschwenderisch!



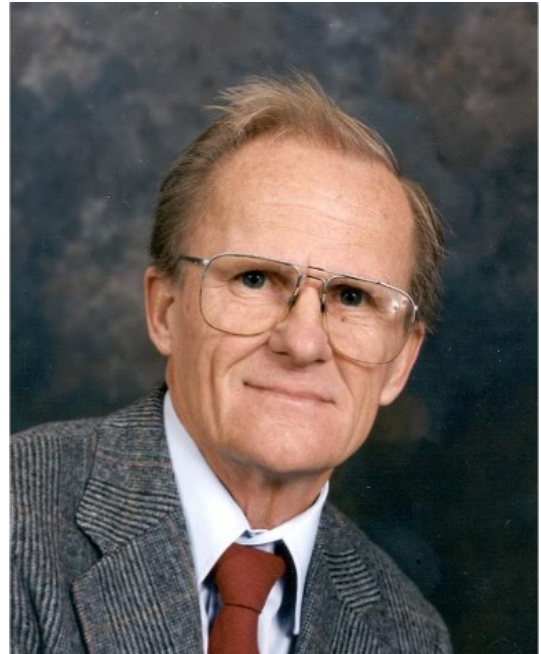
Jorma Rissanen

- Ganze Texte anstatt einzelne Buchstaben kodieren!



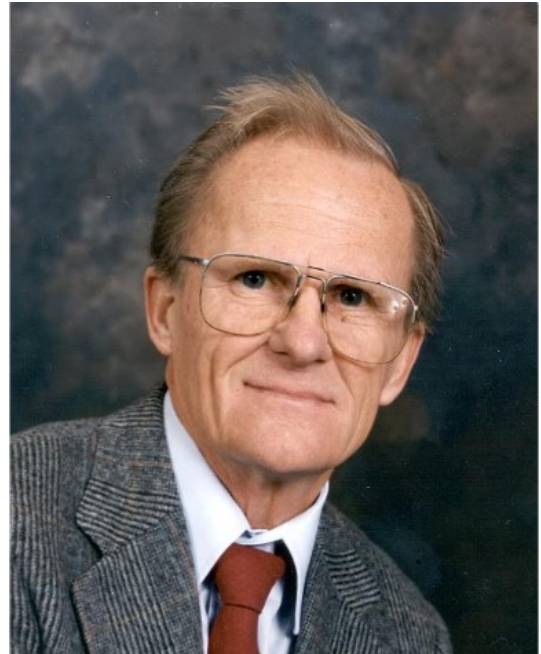
Jorma Rissanen

- Ganze Texte anstatt einzelne Buchstaben kodieren!
- Idee: Jedem Text wird eine Zahl zwischen 0 und 1 zugeordnet



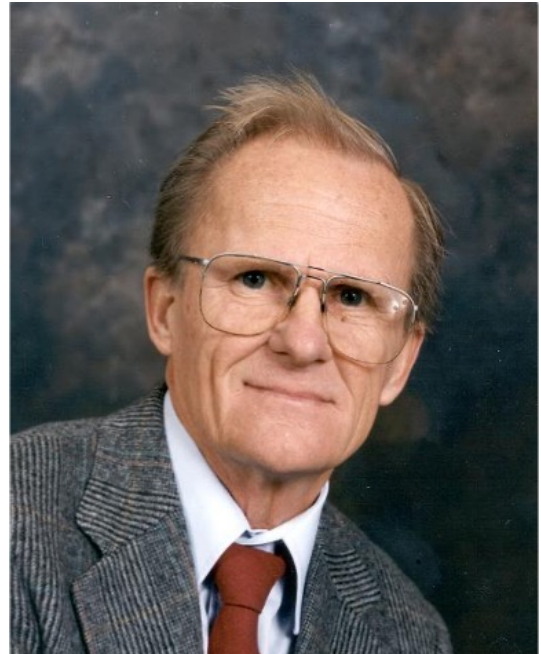
Jorma Rissanen

- Ganze Texte anstatt einzelne Buchstaben kodieren!
- Idee: Jedem Text wird eine Zahl zwischen 0 und 1 zugeordnet
- Je länger der Text, desto mehr Nachkommastellen



Jorma Rissanen

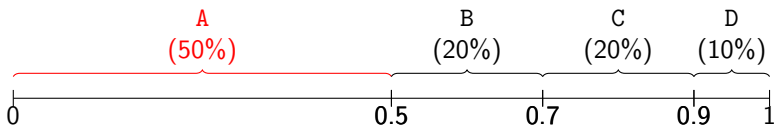
- Ganze Texte anstatt einzelne Buchstaben kodieren!
- Idee: Jedem Text wird eine Zahl zwischen 0 und 1 zugeordnet
- Je länger der Text, desto mehr Nachkommastellen
- Ist unter gewissen Umständen kürzer als Huffman



Arithmetische Kompression

Buchstabe	A	B	C	D
Häufigkeit (%)	50	20	20	10

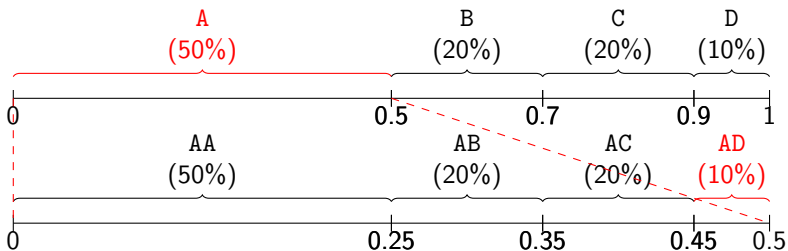
Wie wird der Text `ADBA` kodiert?



Arithmetische Kompression

Buchstabe	A	B	C	D
Häufigkeit (%)	50	20	20	10

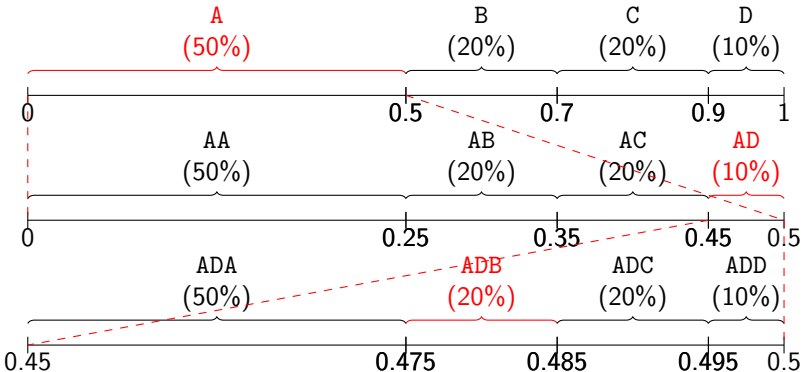
Wie wird der Text ADBA kodiert?



Arithmetische Kompression

Buchstabe	A	B	C	D
Häufigkeit (%)	50	20	20	10

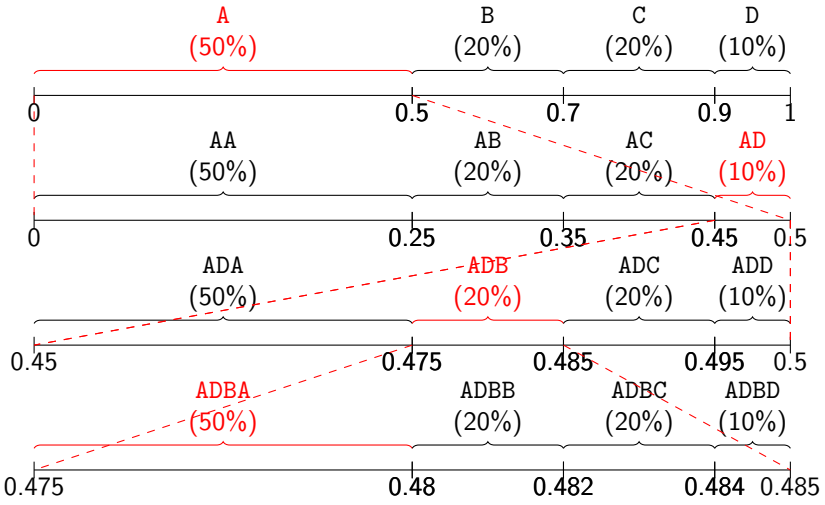
Wie wird der Text ADBA kodiert?



Arithmetische Kompression

Buchstabe	A	B	C	D
Häufigkeit (%)	50	20	20	10

Wie wird der Text ADBA kodiert?



Nachteil von arithmetischer Kodierung

Komprimieren und Dekomprimieren ist mit Huffman effizienter $\rightarrow$ Abwägen (*trade-off*) zwischen Speicherplatz und Rechenaufwand

Auftrag

- Aufgaben 1.10, 1.11
- Challenge: Aufgabe 1.12