

Koding av Unicode-tabellen.

Unicode-tabellen er så lang at vi trenger minst 3 bytes for å lagre hver tallkode. Kjedelig hvis vi bare bruker tegn først i tabellen (og bare noen tegn). UTF-8, UTF-16 er kompakte måter å lagre kodene på som ordner dette.

UTF-8. En kode kan bestå av 1, 2, 3 eller 4 bytes. Tegnene i ASCII-tabellen er de eneste som bare krever en byte.
Se fact 4.19 i Komp.

Komprimering -

Potensielle for komprimering:

Bok: 300 ord pr. side, 4 tegn pr. ord
300 sider, 1 byte pr. tegn:

$1 \cdot 4 \cdot 300 \cdot 300 = 600\,000$ tegn i bok
+ 600 000 bytes med formattering
-tilsammen 700 000 bytes: 700 KB = 0.7 MB

Lyd: Digital lyd basert på 44100 målinger
pr. sekund, i stereo, hver måling lagres
som et heltall med 20 bytes

1 sekund: $44100 \cdot 2 \cdot 2 = 176\,400$ bytes = 176 KB

1 min: 10 MB, 4 min: 40 MB

CD - 1 time: 600 MB.

Video: Enkelt format (PAL): 576 · 720 punkter
3 bytes i hvert punkt, 25 bilder i sekundet.

Et bilde: 1.2 MB, 1 sekund med video: 31 MB

1900 MB pr. min.

112 GB pr. time.

Kompresjon Kap. 7.

Vi har gitt en tekst $x = \{x_1, x_2, \dots, x_m\}$
der hver x_i er hentet fra et alfabet
 $A = \{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n\}$,

$f(\alpha_i)$ - antall forekomster av α_i i x .

$c(\alpha_i)$ - koden til α_i (bit sekvens)

Lagrer x ved å erstatte hver x_i
med $c(x_i)$, dette gir en total bit følge
 z .

Ex. 7.2. $x = DBACDBD$, $A = \{A, B, C, D\}$
 $f(A) = 1$, $f(B) = 2$, $f(C) = 1$, $f(D) = 3$

Vi bruker kodene $c(D) = 0$, $c(B) = 1$, $c(C) = 01$,
Lagrer som z :
 $z = 011001010$ - 9 bits. $c(A) = 10$

Ex. 7.4 Samme tekst med kodene $c(\alpha_i)$ nå:

$c(D) = 1$, $c(B) = 01$, $c(C) = 001$, $c(A) = 000$

Da blir $z = 1010000011011$ - 13 bits.
DBACDBD