

## Huffman - kódolás:

- veszteségmentes tömörítési algoritmus
- titkosításra is használható (Huffman - fa nélkül nem lefordítható vissza)

Tekintsük a következő szöveget: KAKUKKMAPARAMNAK  
16 karakter

1 karakter = 8 bit (ASCII), azaz a szöveg  $8 \cdot 16 = 128$  bit.

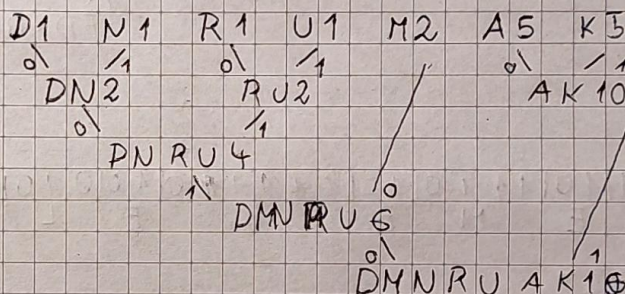
Naiv - tömörítés: csak 7 különböző karakter van, ezért elég 3 bit a karakterek ábrázolásához, azaz a szöveg ellen  $3 \cdot 16 = 48$  biten.

A Huffman - kód a Naiv eszén ötlet fejlettebb változata.

1. lépés gyakorisági táblázatot készítünk a szöveg karaktereiből

A	5
D	1
K	5
M	2
N	1
R	1
U	1

2. lépés a gyakoriság alapján egy minimum prioritásos sorba helyezzük a karaktereket



3. lépés a prioritásos sor elemeiből fát építünk úgy, hogy mindig a két minimumból készítünk csomót

4. lépés megjelöljük az éleket 0 - ha a kisebb gyerek 1 - ha a nagyobb

5. lépés a fából navigálva megkapjuk a karakterek kódjait

A	5	10	5 · 2	} 40 bit
D	1	0100	1 · 4	
K	5	11	5 · 2	
M	2	00	2 · 2	
R	1	0110	1 · 4	
U	1	0111	1 · 4	
N	1	0101	1 · 4	

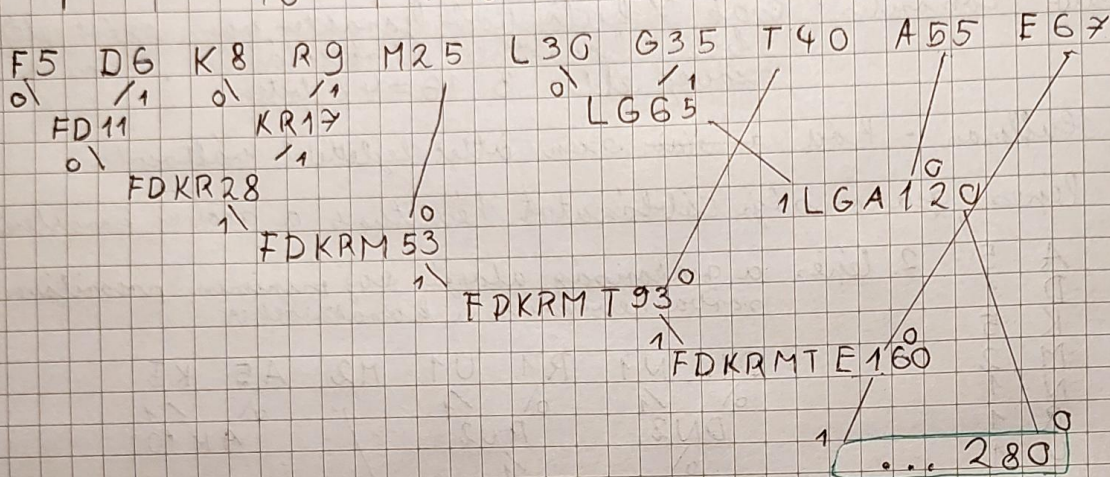
kód méret



# Szorgalmi Huffman - kód

280 exxer Unicode (tipikusan 16 bit) karakter = 4.480.000 bit

Karakter	Gyakoriság	Kód	Méret	
A	55	00	$2 \cdot 55 = 110$	827.000 bit
D	6	111101	$6 \cdot 6 = 36$	<del>Huffman</del>
E	67	10	$2 \cdot 67 = 134$	Huffman - kódolás
F	5	111100	$6 \cdot 5 = 30$	
G	35	011	$3 \cdot 35 = 105$	
K	8	111110	$6 \cdot 8 = 48$	
L	30	010	$3 \cdot 30 = 90$	
M	25	1110	$4 \cdot 25 = 100$	
R	9	111111	$6 \cdot 9 = 54$	
T	40	110	$3 \cdot 40 = 120$	



1111111011101011111010010110  
R E M E K E L T

REMEKELT