

CODIFICAÇÃO DE HUFFMAN

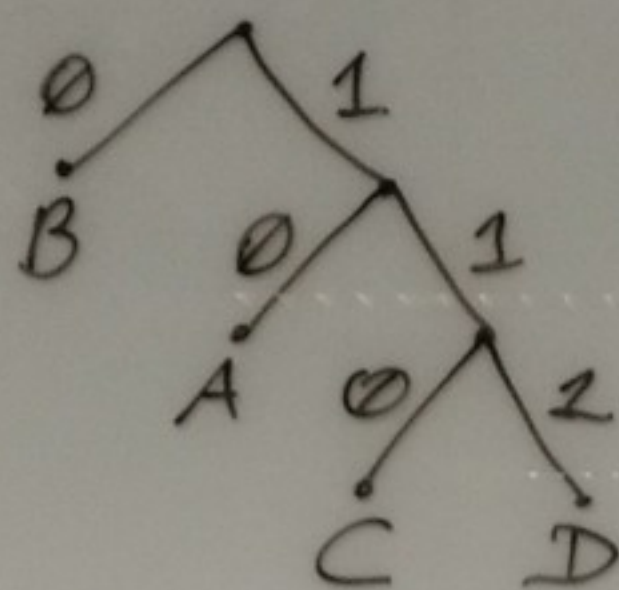
1. INTRODUÇÃO:

a) QUERO ARMAZENAR "ABCDDBB"

A → 00, B → 01, C → 10, D → 11.

TEXTO EM BITS: [000110110101]

12 BITS



b) OUTRA CODIFICAÇÃO:

B: 0

A: 10

C: 110

D: 111

"ABCDDBB"

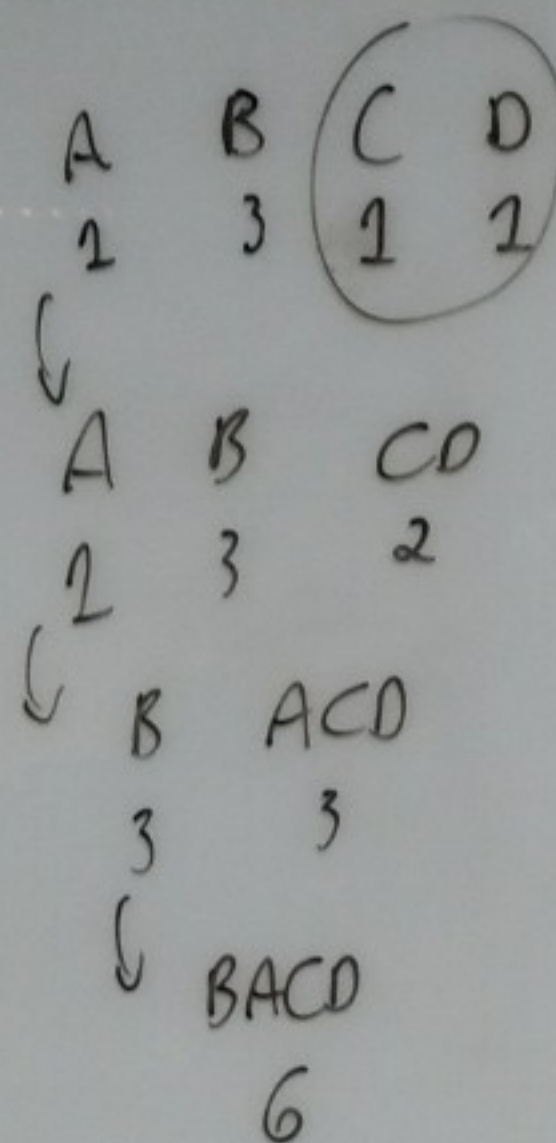
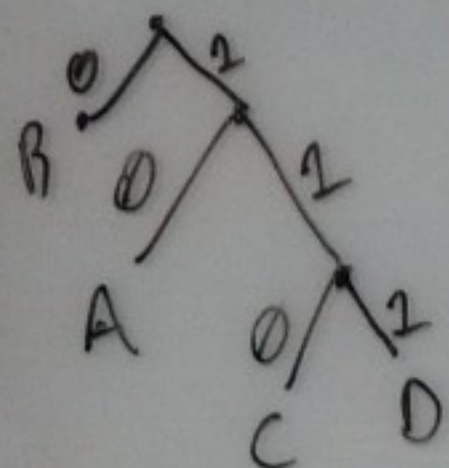
[10011011100]

11 BITS

c) IREMOS ESTUDAR UM ALGORITMO QUE, COM BASE NUM TEXTO FIXO, ESCOLHE A CODIFICAÇÃO LIVRE DE PREFIXOS QUE LEVA AO MENOR TAMANHO POSSÍVEL DO TEXTO CODIFICADO.

2. Codificação de Huffman:

CARACTERE	A	B	C	D
OCCORRÊNCIAS	1	3	1	1



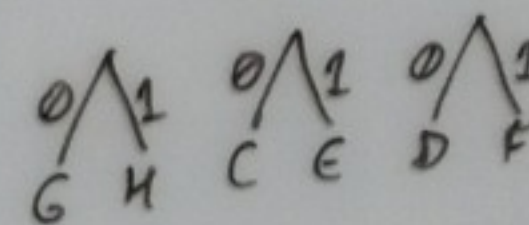
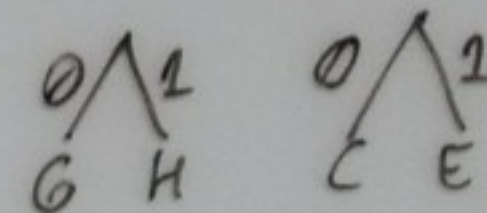
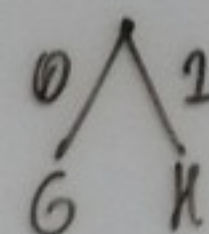
3. Exemplo: "AAAABBBBCCDEFGH"

A	B	C	D	E	F	G	H
4	4	1	1	1	1	1	1

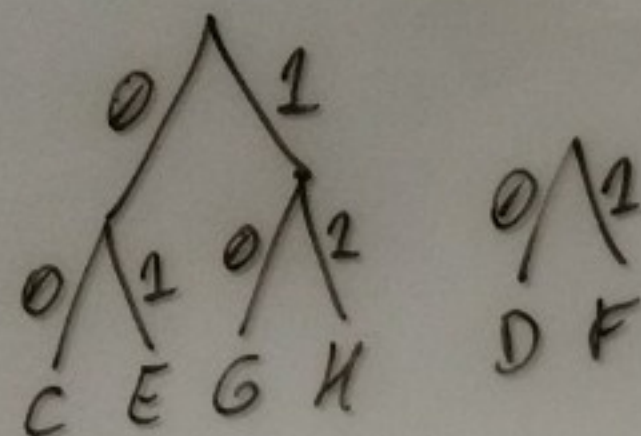
A	B	C	D	E	F	G	H
4	4	1	1	2	1	2	

A	B	CE	D	F	G	H
4	4	2	1	1	2	

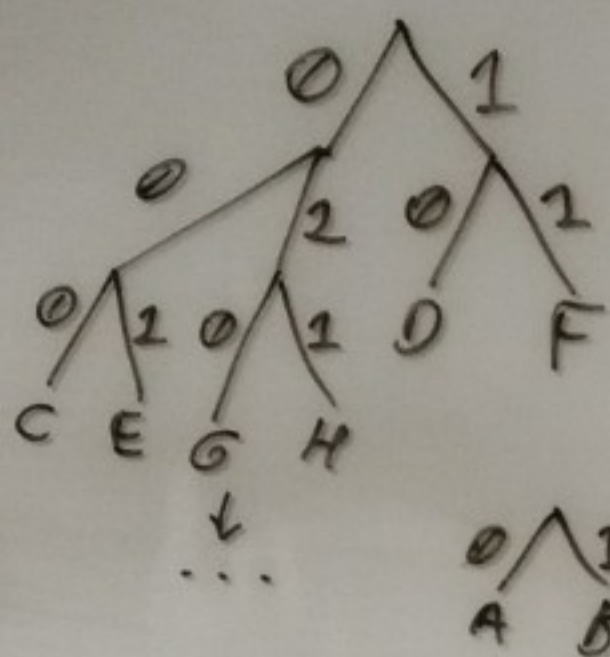
A	B	CE	DF	G	H
4	4	2	2	2	



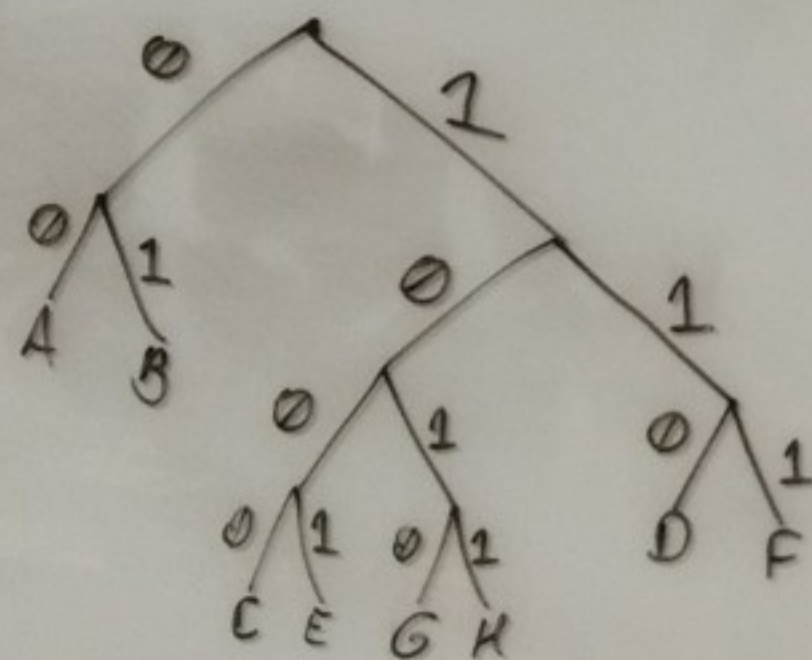
↓ A B C E G H D F
4 4 4 2



↓ A B C E G H D F
4 4 6



↓ AB C E G H D F
8 6
↓ ABC E G H D F
24



TAMANHO-DO-TEXTO-COMPACTADO =

$$\sum_{l \in A} OC[l] \cdot TAM_CÓD(l)$$

$$= \underbrace{4 \cdot 2}_A + \underbrace{4 \cdot 2}_B + \underbrace{1 \cdot 4}_C + \underbrace{1 \cdot 3}_D + \underbrace{1 \cdot 4}_E +$$

$$+ \underbrace{1 \cdot 3}_F + \underbrace{1 \cdot 4}_G + \underbrace{1 \cdot 4}_H$$

$$= 8 + 8 + 4 + 3 + 4 + 3 + 4 + 4$$

$$= \underline{38 \text{ bits.}}$$

Com códigos de mesmo tamanho, o tamanho é $14 \times 3 = \underline{42 \text{ bits.}}$

4. HUFFMAN EM PSEUDOCÓDIGO:

ALGORITMO: HUFFMAN

ENTRADA: VETOR $O[0..n-1]$ // OCORRÊNCIAS

SAÍDA: ÁRVORE BINÁRIA

$H :=$ HEAP VAZIO (DE MÍNIMO)

PARA i DE 0 A $n-1$

[INSIRA i EM H COM PESO $O[i]$]

ENQUANTO $|H| \neq 1$

REMOVA O MÍNIMO a DE H , DE PESO p_a

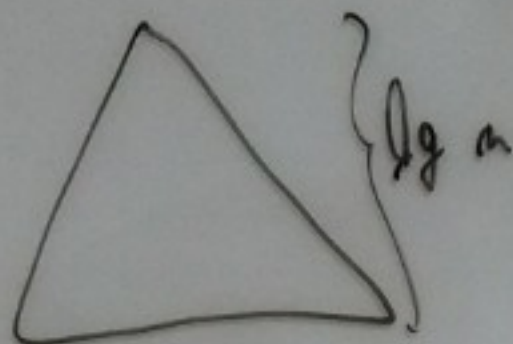
REMOVA O MÍNIMO b DE H , DE PESO p_b

$c :=$ NOVO NÓ

FAÇA a FILHO ESQ. DE c , e b FILHO DIR.

INSIRA c EM H COM PESO $p_a + p_b$

RETORNE O ELEMENTO ÚNICO DE H .



5. RECAPITULANDO HEAPS (HEAP DE MÍNIMO):

