

STATISTICAL CODING : based on frequency of bits/bytes/words, integers

- Huffman
- Arithmetic

Frequency: STATIC MODE $\downarrow$  are given  
 SEMI STATIC MODE $\downarrow$  of frequency  
 DYNAMIC MODE $\downarrow$  SCAN the text, count of chars

$\Sigma$ : alphabet of symbols that we will encode, binary, bytes, words, integers  
 $(\infty)$   
 $(2)$   
 $(256)$   
 $(\text{millions of symbols})$

Compressed file:

$C(t) =$	PREAMBLE	Body
	$\downarrow$	$\downarrow$
- additional info:		sequence of bits
- $\Sigma$		
- frequency		
...		

## HUFFMAN

greedy algorithm  $\rightarrow$  optimal

the algo repeats these steps:

tree bottom-up

1. pick the 2 lightest (probability) symbols

2. merge them in one new symbol whose prob. is the sum of the probabilities of the merged symbols

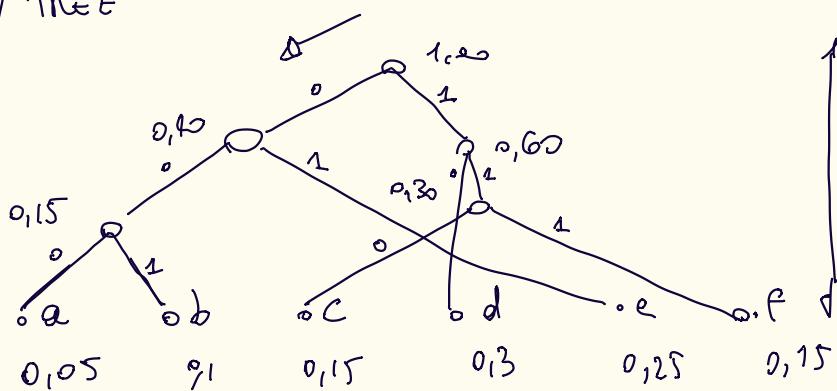
given an alphabet  $\Sigma = \{a, b, c, d, e, f\}$  with frequencies:

$\begin{matrix} & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ a & 0,05 & b & 0,1 & c & 0,15 & d & 0,3 & e & 0,25 & f & 0,15 \end{matrix}$

$T = a a a b b b$

construct a BINARY TREE

generates codewords  
 $\xrightarrow{\text{from root}}$



a: 000  
 b: 001  
 c: 110  
 d: 10  
 e: 01  
 f: 111

PROPERTIES

① is a PREFIX-FREE CODE: no codeword is a prefix of the other

② common rules of data compression: most frequent symbols get the shortest code  
 $\hookrightarrow$  merges light symbols first

To evaluate the goodness of a code:  $\text{AVG codeword length } L_c = \sum_{c \in \Sigma} f(c) \cdot l(c)$ .  $f(c)$  is the freq. of the symbol,  $l(c)$  is the length of the codeword.

$$L_c = 3 \cdot 0,05 + 3 \cdot 0,1 + 3 \cdot 0,15 + 2 \cdot 0,3 + 2 \cdot 0,25 + 3 \cdot 0,15 = 3 \cdot 0,45 + 2 \cdot 0,55 = 1,35 + 1,1 = 2,45 \text{ bits}$$

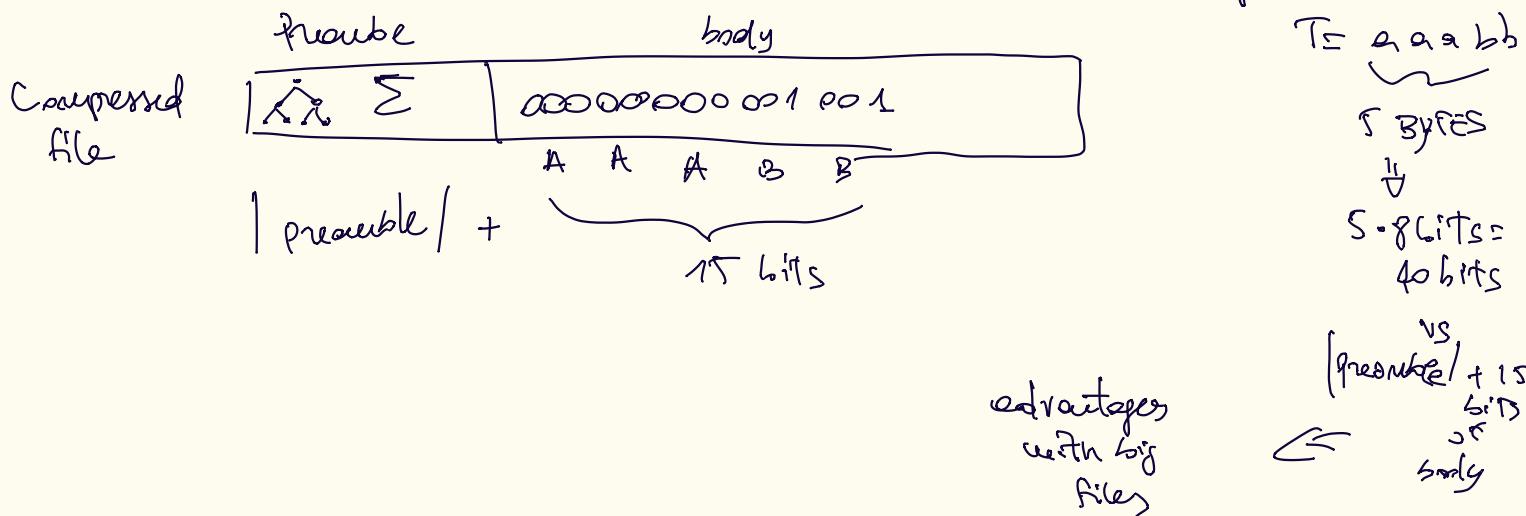
$\Rightarrow$  Avg cw length of the code

Relation between:

Avg cw length & height of the tree

$\Rightarrow$  height of the tree = cw length

$\Rightarrow$  Avg height of the tree = Avg cw length ( $L$ )



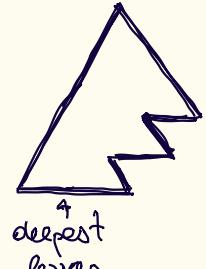
Theo: Huffman code is optimal among prefix-free codes for  $\Sigma$ .  
 (Take a symbol, assign a codeword)

Avg length of Huffman:  $L_H \leq L_c$  Avg length of prefix-free code if code  $C$  prefix-free

Lemma: let  $F$  be the set of binary trees corresponding to optimal prefix-free codes

## CANONICAL HUFFMAN:

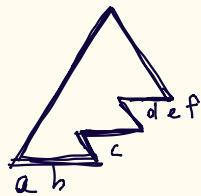
- 1) Reduce space of Huffman tree
- 2) Reduce compression time

WHAT that  
 $\Rightarrow$  Huffman  
 tree gets  
 this form:  


1. Compute the codeword length  $L(\sigma)$  for each symbol  $\sigma \in \Sigma$  according to the classical **Huffman's algorithm**. Say  $\max$  is the maximum codeword length.
2. Construct the array  $\text{num}[1, \max]$  which stores in the entry  $\text{num}[\ell]$  the number of symbols having Huffman codeword of  $\ell$ -bits.
3. Construct the array  $\text{symb}$  which stores in the entry  $\text{symb}[\ell]$  the list of symbols having Huffman codeword of  $\ell$ -bits.
4. Construct the array  $\text{fc}[1, \max]$  which stores in the entry  $\text{fc}[\ell]$  the first codeword of all symbols encoded with  $\ell$  bits;
5. Assign consecutive codewords to the symbols in  $\text{symb}[\ell]$ , starting from the codeword  $\text{fc}[\ell]$ .

so

- 0) Build Huffman tree



- 1) Compute cw lengths:

	cw length
a	4
b	4
c	3
d	2
e	2
f	2

- 2) Compute # of symbols per level length (Num array)

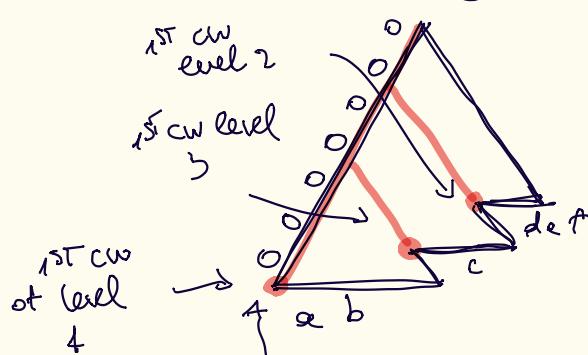
num	1	2	3	4
	0	3	1	2

- 3) Compute symbol table: for every level we keep the list of the symbols at that level

Symbol	0	1	2	3	...
1					
2		d, e, f			
3			c		
4				a, b	

- 4) Compute the new codewords  
 Create the tree here)

observe that the tree has a special shape: in order to describe the structure of the tree: I need the 1<sup>st</sup> codeword of every level



1<sup>st</sup> cw: f<sub>c</sub>

FORMULA

-  $l = \text{level max}$  (in our case)

-  $f_c[l_{\max}] = 0$

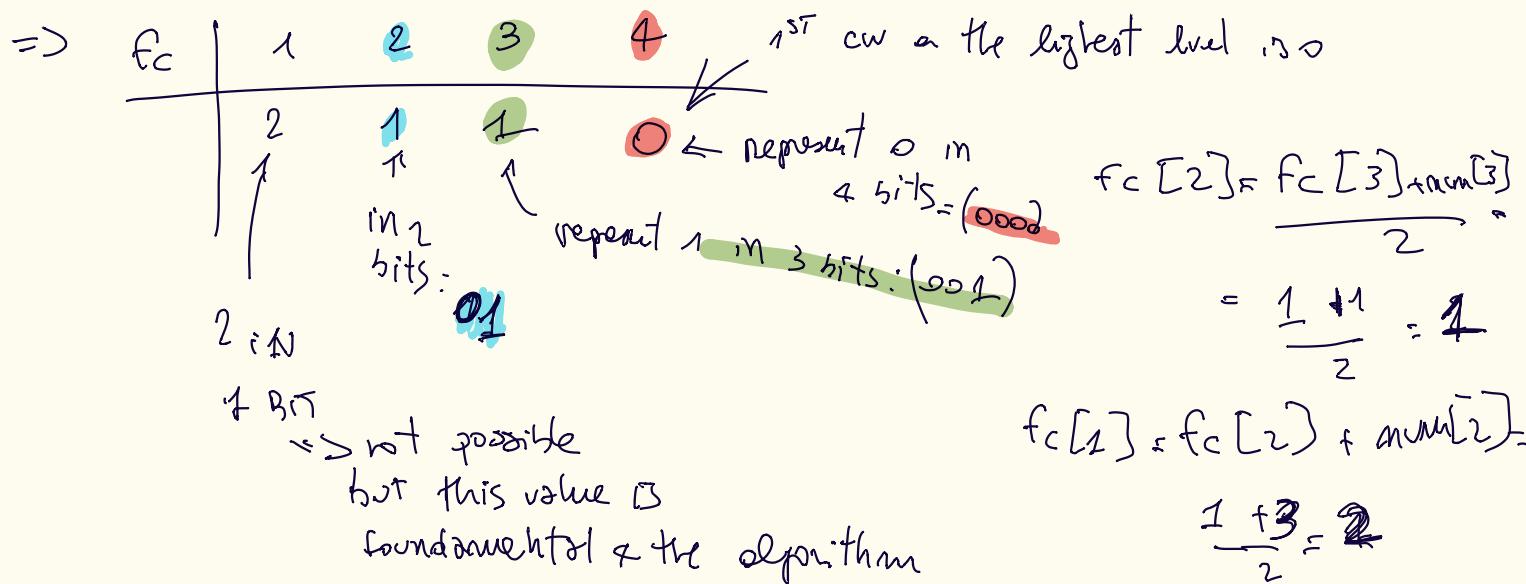
$f_c[l] = f_c[l+1] + \frac{\text{num}[l+1]}{2}$

$$\text{next level } \frac{\text{num}[f_c[l+1]]}{2} \text{ at the next level}$$

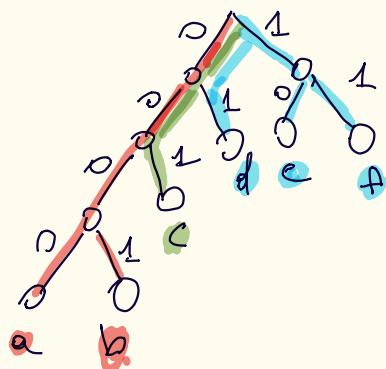
- if I've  $f_c$  (1<sup>st</sup> codeword)  
 the others are obvious w/ others  
 are just increasing
- and the symbols are in the  
 symbol table

Ex. for level 3:

$$f_c[3] = \frac{f_c[4] + \text{num}[4]}{2} = \frac{0+2}{2} = 1$$



$\Rightarrow$



$\Rightarrow$  Canonical Huffman tree  
we store f<sub>c</sub>[L], num  
Symb. table

How we code: T = b c f

b is in position 1 in the symbol table and is related to level 4

$\Rightarrow$  encode b: f<sub>c</sub>[4] + rank(b) in its list

$\underbrace{0}_{0} + \underbrace{1}_{1}$

cw = 1 represented in 4 bits: 0001

c: f<sub>c</sub>[3] + rank(c)

$\underbrace{1}_{1} + \underbrace{0}_{0}$

cw 1 in 3 bits: 001

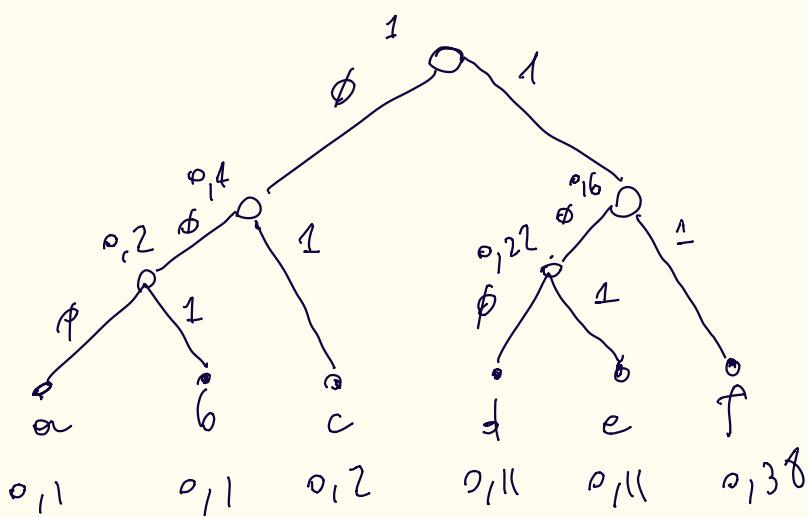
f: f<sub>c</sub>[2] + rank[f]

$\underbrace{2}_{2} + \underbrace{2}_{2}$

es: Canonical Huffman:

S: a b c d e f  
 $\begin{matrix} 0,1 & 0,1 & 0,2 & 0,11 & 0,11 & 0,38 \end{matrix}$

a encode  $T = f A D E$



o) compute cw length

$$a = 000 \quad 3$$

$$b = 001 \quad 3$$

$$c = 01 \quad 2$$

$$d = 100 \quad 3$$

$$e = 101 \quad 3$$

$$f = 11 \quad 2$$

f) compute num. array  
 num. array

num	1	2	3
num	0	2	4

symbol table

symbol	0	1	2	3
'a'				1
'b'			1	2
'c'		1	2	3

3) compute 1st cw length

$$f_c[l_{\max}] :=$$

$$f_c[l] = \frac{f_c[l+1] + \text{num}[l+1]}{2} :=$$

$$f_c[2] = \frac{f_c[3] + \text{num}[3]}{2} :=$$

$$= \frac{0 + 4}{2} = 2$$

$$f_c[1] = \frac{f_c[2] + \text{num}[2]}{2} :=$$

$$= \frac{2 + 2}{2} = 2$$

$$T = f_c[\text{level}] + \text{rank in symbol table} :=$$

$$= f_c[2] + \text{rank}[f] =$$

$$2 + 1 = 3 \text{ in } 2 \text{ bits} \Rightarrow 11$$

$$A = f_c[3] + \text{rank}[A] =$$

$$= 0 + 0 = 0 \text{ in } 3 \text{ bits} \Rightarrow \infty$$

$$D = f_C[3] + \text{rank}[d] =$$

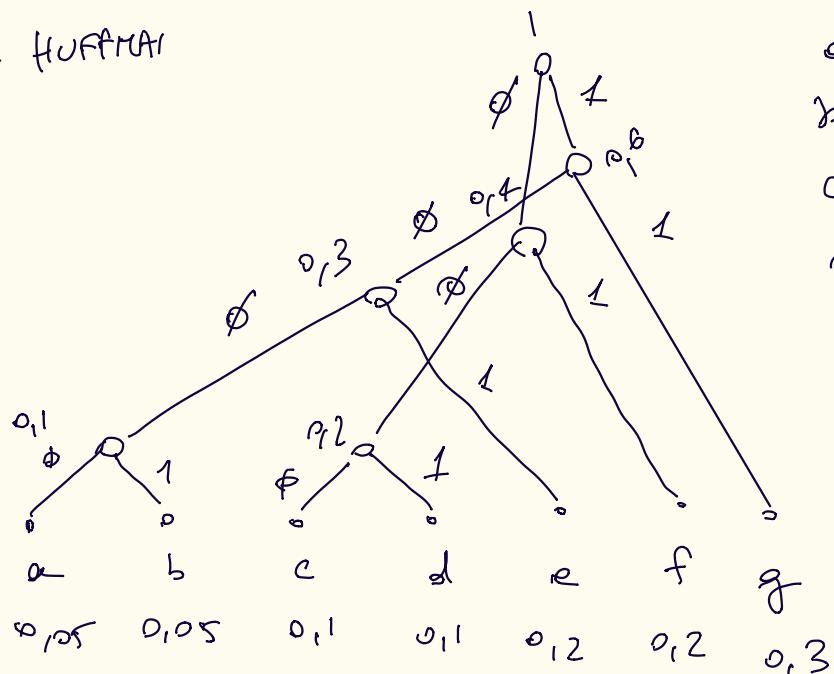
$$0 + 2 = 2 \text{ in } 3 \text{ bits} \Rightarrow 010$$

$$E = f_C[3] + \text{rank}[E] =$$

$$0 + 3 = 3 \text{ in } 3 \text{ bits} \Rightarrow 011$$

(✓)

CANONICAL HUFFMAN



	code length
a = 1000	4
b = 1001	4
c = 000	3
d = 001	3
e = 101	3
f = 01	2
g = 11	2

num array

.	1	2	3	4
num	0	2	3	2

Symbol table

symbol	0	1	2	3
1				
2			f, g	
3			D, E	
4	A, B			

compute  $F_C$  per level

e	1	2	3	4
FC	2	2	1	0

$$F_C[l_{\max}] = 0$$

$$F_C[l] = F_C[l+1] + \frac{\text{num}[l+1]}{2}$$

$$F_C[3] = \frac{F_C[4] + \text{num}[4]}{2} =$$

$$F_C[2] = \frac{F_C[3] + \frac{4}{2}}{2} = \frac{0 + 2}{2} = 1$$

$$F_C[1] = \frac{2 + 2}{2} = \frac{4}{2} = 2$$

encode  $\tau = G \subset D$

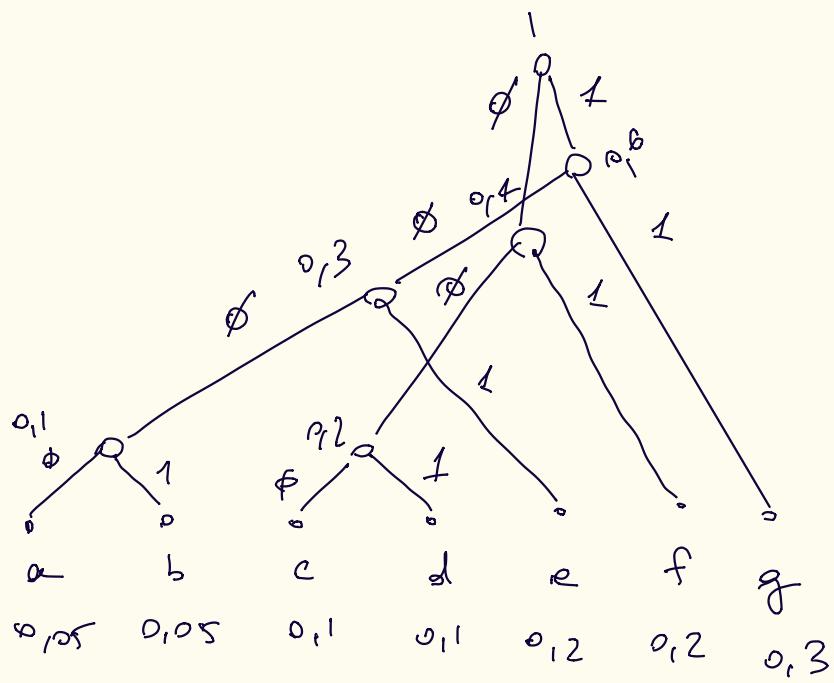
$$G = FC[\text{level}] + \text{rank}[\underline{\text{G}}] =$$

$$FC[2] + 1 = 2 + 1 = 3 \text{ in } 2 \text{ bits} \Rightarrow 11$$

$$C = FC[3] + \text{rank}[C] = 1 + 0 = 1 \text{ in } 1 \text{ bit} \Rightarrow 0$$

$$D = FC[3] + \text{rank}[D] = 1 + 1 = 2 \text{ in } 3 \text{ bits} \Rightarrow$$

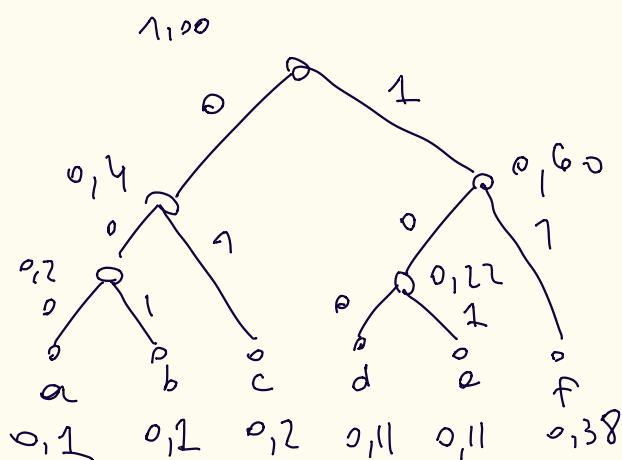
010



CANONICAL Huffman

S: a b c d e f  
 $\omega_1$  0,1 0,1 0,2 0,11 0,11 0,38

a encode  $T = f A \triangleright E$



A = 000  
 B = 001  
 C = 01  
 D = 100  
 E = 101  
 F = 11

NUM	1	2	3
	0	2	4

SYMB	0	1	2	3
1	C	F		
2	A	B	D	E

FC	1	2	3
	2	2	0

$$l_{\max} = 0$$

$$FC[l] = \underbrace{FC[l+1]}_{l=2} + \underbrace{NUM[l+1]}_{l=2}$$

$$FC[2] = \frac{0+4}{2} = \frac{4}{2} = 2$$

$$FC[1] = \frac{2+2}{2} = \frac{4}{2} = 2$$

ENCODE:  $FC[l] + \text{RANK}[symbol]$

$$F = FC[2] + \text{RANK}[f] = 2 + 1 = 3 \text{ in } 2 \text{ bits } (3)_{10} = (11)_2$$

$$A = FC[3] + \text{RANK}[A] = 0 + 0 = 0 \text{ in } 3 \text{ bits } (0)_{10} = (000)_2$$

$$D = FC[3] + \text{RANK}[D] = 0 + 2 = 2 \text{ in } 3 \text{ bits } (2)_{10} = (010)_2$$

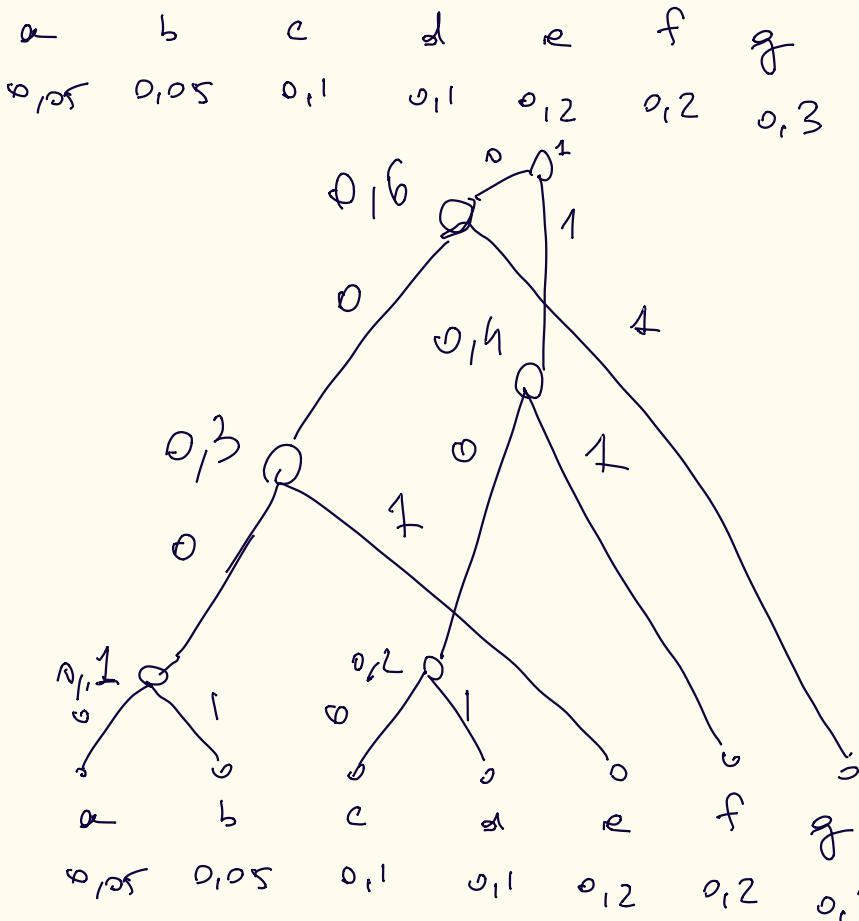
$$E = FC[3] + \text{RANK}[E] = 0 + 3 = 3 \text{ in } 3 \text{ bits } (3)_{10} = (011)_2$$

DECODE: 11000 010 011  
 $\begin{matrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{matrix}$

$V=1 \quad l=1 \quad V \leq 1 \quad ? \quad FC[1] \text{ yes} \Rightarrow V = 2 \cdot 1 + 1 = 3 \quad l+1$

$v = 3 \quad l = 2$	$v = 3 \quad ? \quad FC[2] = 2 \quad NO$	$\text{not symbol}[l; v - FC[l]] =$
$v = 0 \quad l = 1$	$v = 0 \quad ? \quad FC[1] = 2 \quad YES$	$= \text{symbol}[l; 3 - 2] = [1, 1]$
$v = 0 \quad l = 2$	$v = 0 \quad ? \quad FC[2] = 2 \quad YES$	$v = 2 \cdot 0 + 0 = 0 \quad l++ \quad = \textcircled{F}$
$v = 0 \quad l = 3$	$v = 0 \quad ? \quad FC[3] = 0 \quad NO$	$v = 2 \cdot 0 + 0 = 0 \Rightarrow l++$
$v = 0 \quad l = 4$		$\text{symbol}[l; v - FC[l]] = \text{sy}[3; 0 - 2] = \textcircled{A}$
$v = 1 \quad l = 2$	$v = 0 \quad ? \quad FC[2] = 2 \quad YES$	$v = 2 \cdot 0 + 1 = 1 \quad l++$
$v = 2 \quad l = 3$	$v = 1 \quad ? \quad FC[3] = 0 \quad NO$	$v = 2 \cdot 1 + 0 = 2 \quad l++$
$v = 0 \quad l = 1$	$v = 0 \quad ? \quad FC[1] = 1 \quad NO$	$\text{not symbol}[3; 2 - 2] = \textcircled{D}$
$v = 1 \quad l = 2$	$v = 1 \quad ? \quad FC[2] = 2 \quad NO$	$v = 2 \cdot 0 + 1 = 1 \quad l++$
$v = 3 \quad l = 3$	$v = 1 \quad ? \quad FC[3] = 0 \quad NO$	$v = 2 \cdot 1 + 1 = 3 \quad v++$
		$\text{symbol}[3; 3 - 0] = \textcircled{O}$

# Do Canonical Huffman



Code Table:

a	0000
b	0001
c	110
d	111
e	001
f	11
g	01

NUM	1	2	3	4
	0	2	3	2

Symb	0	1	2	3
1	-			
2	F	G		
3	C	D	E	
4	A	B		

FC	1	2	3	4
	2	2	1	0

$$FC[l] = \frac{FC[l+1] + \text{numu}[l+1]}{2}$$

$$FC[3] = \frac{FC[4] + \text{numu}[4]}{2} = \frac{0 + 2}{2} = 1$$

$$FC[2] = \frac{1 + 3}{2} = \frac{4}{2} = 2$$

$$FC[1] = \frac{2 + 2}{2} = \frac{4}{2} = 2$$

ENCODE :  $FC[l]$  + rank [ $G^{n \text{ symb}}$ ] in  $\hat{F}_c$  bits

$T = G \leftarrow D$

$G = FC[2] + \text{RANK}[G.] = 2 + 1 = 3$  in 2 bits

3 bits  $\Rightarrow 011$

$C = FC[3] + \text{RANK}[C] : 1 + 0 = 1$  in 3 bits

$D = FC[3] + \text{RANK}[D] = 1 + 1 = 2$  in 3 bits  $\Rightarrow 001$

010

DECODE : 011001010  
          ↑↑↑

$V = \text{next\_bit}()$

$l = 1;$

$V < FC[l]$  yes  $\Rightarrow$  continue

$V = 2V + \text{next\_bit}$

$l++$

else

decode  $\Rightarrow$  return symbols [ $l$ ;  $V - FC[l]$ ]

$V = 0 \quad l = 1$

$0 < FC[1]$  yes  $\Rightarrow V = 2 \cdot 0 + 1 = 1 \quad l = 2$

$1 < FC[2]$  yes  $V = 2 \cdot 1 + 1 = 3 \quad l = 3$

$3 < FC[3]$  no  $3 ; 3 - 1 = 2$   
return symbols [ $l$ ;  $V - FC[l]$ ]  
 $symbols[3; 2]$

$symbols[3; 2]$

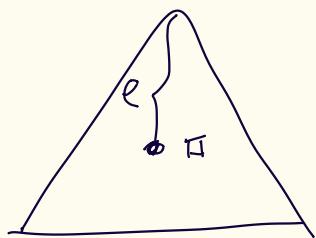
# DECODING

$f_c$	1	2	3	4
	2	1	1	0

symbol table

symbol	0	1	2	3
1				
2	c, f			
3	a, b, d, e			

. we need  $\ell$ , called  $v$



$v$

1 0 0 0 1 0 1 0

c

if ( $v < f_c[\ell]$ ) go down

ALGORITHM

fetch 1st bit  
start with level 1  
 $v = \text{next\_bit}(); \quad \ell = 1;$  1st cw of  $\ell$   
while ( $v < f_c[\ell]$ )  
 $v = 2 \cdot v + \text{next\_bit}()$   
 $\ell++;$   
return symbol [ $\ell, v - f_c[\ell]$ ])

else decode

$v$

$2^{\ell} v$  (add 0 in this pos)

next bit → if it's 0 remains 0  
if 1 get 2

// go down

$\Rightarrow$  Decode  $S = 11001$

$v = 1; \quad \ell = 1;$

$f_c$	1	2	3	4
	2	1	1	0

symbol table

symbol	0	1	2	3
1	"			
2	d	e	f	
3	c			
4	a	b		

$v = 1 \nless f_c[1] = 2$  yes continue

$v = 2 \cdot v + \text{next\_bit}() = 2 \cdot 1 + 1 = 3; \quad \ell = 2;$

$v = 3 \nless f_c[2] = 1$  no  $\Rightarrow$  decode

$\Rightarrow$  return symbol [ $\ell, v - f_c[\ell]$ ])

$\Rightarrow$  symbol [2, 2] = f  $\checkmark$

REASON

$v = 0; \quad \ell = 1;$

$v = 0 \nless f_c[1]$  yes continue

$v = 2 \cdot v + \text{next\_bit}() = 0 + 0 = 0 \quad \ell = 2$

$v=0$ ;  $l=2$   
 $v=0 \geq fc[2]$  yes continue

$$v = 2 \cdot 0 + \text{rex}[6] = 0 + 1 = 1; l=3$$

$v=1 \geq fc[3]$  NO == Decade

$\text{symb}[l, v-fc[l])]$

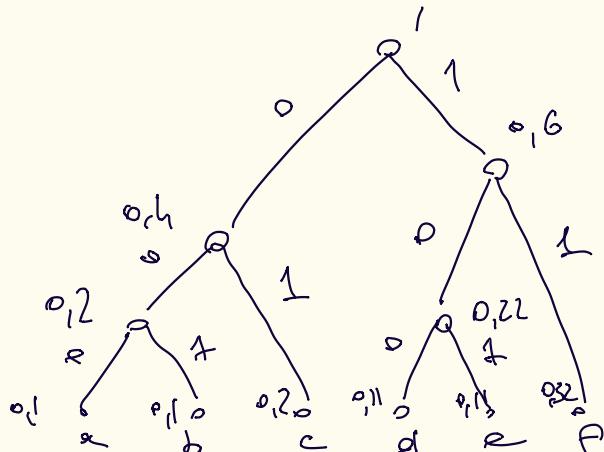
$$\Rightarrow \text{symb}[3, 1-1] = [3, 0] = \text{C}$$



```
v = next_bit();
l = 1;
while( v < fc[l] )
    v = 2v + next_bit();
    l++;
return symb[ l, v-fc[l] ];
```

Canonical Huffman

Symbol	a	b	c	d	e	f
prob.	0,1	0,1	0,2	0,11	0,11	0,32



encode

$$\begin{aligned}
 a &= 000 \\
 b &= 001 \\
 c &= 01 \\
 d &= 100 \\
 e &= 101 \\
 f &= 11
 \end{aligned}$$

num	1	2	3
	0	2	4

symbol	0	1	2	3
1	c			
2		c, f		
3			A, B, D, E	

FC	1	2	3
	2	2	0

$$FC[l_{\max}] = 0$$

$$FC[l] = \frac{FC[l+1] + NUM[l+1]}{2}$$

$$FC[2] = FC[3] + NUM[3] = \frac{0+4}{2} = 2$$

$$FC[1] = \frac{FC[2] + NUM[2]}{2} = \frac{2+2}{2} = 2$$

$$\begin{aligned}
 \text{ENCODE} &= \text{FC}[ \text{Symbol} ] + \text{RANK}[ \text{Symbol} ] \text{ in FC bits} \\
 \text{Symbol} &= \text{FC}[ \text{Symbol} ] + \text{RANK}[ \text{Symbol} ] \text{ in FC bits}
 \end{aligned}$$

$$= \frac{2+2}{2} = 2$$

T = F A D E

$$F = FC[2] + RANK[F] = 2 + 1 = 3 \text{ in } 2 \text{ bits} \Rightarrow 11$$

$$A = FC[3] + RANK[A] = 0 + 0 = 0 \text{ in } 2 \text{ bits} \Rightarrow 00$$

$$D = FC[3] + RANK[D] = 0 + 2 = 2 \text{ in } 2 \text{ bits} \Rightarrow 010$$

$$E = FC[3] + RANK[E] = 0 + 3 = 3 \text{ in } 2 \text{ bits} \Rightarrow 011$$

ପ୍ରକାଶ

```

r = next_bit();
l = 1
while (v < fc[l])
    r = 2 * r + next_bit();
    l++;
return symb[l; v - fc[l]];

```

— C —

Decode first 3 symbols of 10001011011100

10001011  
↑↑↑↑↑↑↑↑↑↑

$$V = \frac{1}{1}, \quad e = 1$$

$$(\zeta_1 < f([e])) \quad si$$

$$V = 2 \cdot 1 + 0$$

$$l=2;$$

$$v=2; \ell=2;$$

"  $(z \in f_C[z])$  no

$\Rightarrow$  return symbol[ $l$ ;  $v - f(l)$ ]

$$\text{Spur}[z_1^{\infty}] \Rightarrow$$

— — —

$$V=0 \quad ; \quad \ell=1$$

$$= 1 \quad (\rho < f([4])) \quad \text{yes}$$

$$v = 2 \cdot 0 + \text{rest\_bit}() = 0 + 0$$

$$\ell = 2$$

$$V=0 \quad \ell=2$$

$(0 < \tau_C[2])$  yes

$$V = 2.0 + \text{rect}_{-1.7} = 1$$

$$v=1 \quad \ell=3$$

(1 <  $F_C$  [3]) ~

return symbol [ e ; r - fc[<sup>''</sup>e] ] >

skew [3; 1-0-1]  $\Rightarrow$  (B)

$v = \text{vert\_bit} := 0$        $\ell = 1$   
 $(0 < FC["\ell"])$  yes

$$v = 2 \cdot 0 + \text{hex}1 \approx 0 + 1 = 1$$
$$\ell++;$$
       $\ell = 2$

$v = 1$      $\ell = 2$   
 $(1 < FC["2"])$  yes

$$v = 2 \cdot 1 + \text{hex}1 = 2 + 1 = 3$$

$\ell++;$

$v = 3$      $\ell = 3$   
 $(3 < FC[3])$  no

$\Rightarrow \text{return symbol}[\ell; v - FC[\ell]]$   
 $\Rightarrow \text{symbol}[3; 3] \Rightarrow \text{(E)}$

Decode first 3 symbols of  $\begin{smallmatrix} 1000101101100 \\ \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \end{smallmatrix}$

$f_C$	1	2	3
	2	2	0

$$v = 1 \quad l = 1$$

$$1 < v \quad ? \quad f_C[l] = 2$$

yes  $\Rightarrow$

$$v = 2 + 0 = 2 \quad l++$$

symbol	0	1	?	3
1	c			
2		c, f		
3	a, b, d, e			

$$v = 2 \quad l = 1$$

$$2 < v \quad f_C[2] = 2 \quad \text{no}$$

$$\text{decode} \Rightarrow \text{return } \text{symbol}[l; v - f_C[l]] = [2; 0]$$

return  $(c)$

$\sim \sim \sim$

$$v = 0 \quad l = 1$$

$$0 < v \quad f_C[1] \Rightarrow v = 2 \cdot 0 + 0 \quad l++ \quad \sim \sim \quad l = 1$$

$$v = 0 \quad l = 2$$

$$0 < v \quad f_C[2] \Rightarrow v = 2 \cdot 0 + 1 \quad l = 2$$

$$l = 3 \quad v = 1$$

$$1 < v \quad f_C[3] \quad \text{no} \Rightarrow \text{decode}$$

$$\text{return } \text{symbol}[l; v - f_C[3]] = [3, 1]$$

$\Rightarrow (B)$

$$v = 0 \quad l = 1$$

$$0 < v \quad f_C[1] \Rightarrow v = 2 \cdot 0 + 1 = 1 \quad l++$$

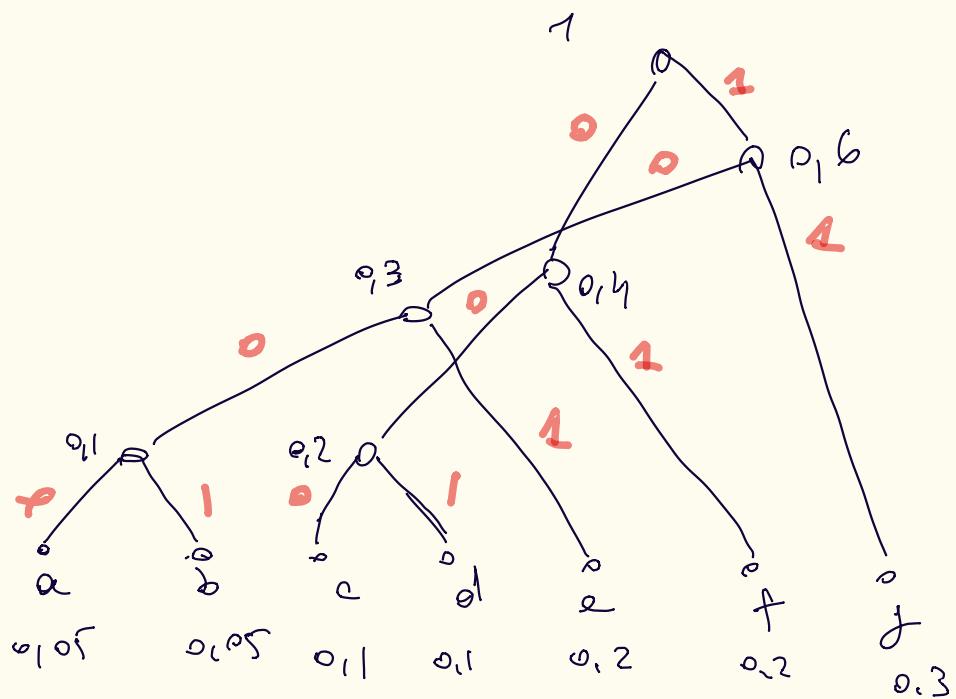
$$v = 1 \quad l = 2 \quad 1 < v \quad f_C[2] \quad v = 2 \cdot 1 + 1 = 3 \quad l++$$

$$v = 3 \quad l = 2$$

$$3 < v \quad f_C[2] \quad \text{no}$$

$$\text{return } \text{symbol}[l; v - f_C[l]] = [2; 1] \Rightarrow (F)$$

## Canonical Huffman



A horizontal number line with tick marks at integer intervals from 1 to 4. Above the line, the numbers 1, 2, 3, and 4 are written. Below the line, the label "NUM" is positioned above the tick mark for 1, and the symbol "≈" is positioned above the tick mark for 2. The tick marks for 3 and 4 are aligned with the numbers 3 and 4 respectively.

<u>Symbols</u>	0	1	2
1	$\approx$		
2	F	G	
3	C	D	E
4	A	B	

$\mathcal{F} \subset [l_{\max}]^{\leq}$

$$FC[l] = FC[l+1] + \text{num}[l+1]$$

$$FC[3] = \frac{FC[4] + NUM[4]}{2} \leftarrow 0 + 2 = 1$$

$$FC[2] = \frac{FC[3] + NM[3]}{2} = \frac{1+3}{2} = 2$$

$$FC[1] = \frac{FC[2] + NUM[2]}{2} = \frac{2+2}{2} = 2$$

ENCODE: f symb

SYMB = FCL[ l ] + rank[symb]

$$G = \text{fc}[2] \text{ rank}[G] = 2+1=3 \text{ in } 2 \text{ bits}$$

11

$C = FC[3] + \text{rank}[C] = 1 + 0 = 1$  in 3 bits  $\Rightarrow$  (001)

$D = FC[3] + \text{rank}[D] = 1 + 1 = 2$  in 3 bits  $\Rightarrow$  (010)

Decode with canonical Huffman, 1st 3 symbols

ACG  
 000 10 11 01 00 10  
 111 ↑↑↑ 111 111 ↑  
 A C G

$v = \text{next\_bit}; l = 1;$

while ( $v < FC[l]$ )

$v = 2 \cdot v + \text{next\_bit}();$   
 $l++;$

return symbol[l; v - FC[l]]

→ → →

$v = 0 \quad l = 1$

$0 < FC[1] \Rightarrow v = 2 \cdot 0 + 0 = 0$   
 $l++$

$v = 0 \quad l = 2$

$0 < FC[2] \quad \text{yes} \Rightarrow v = 2 \cdot 0 + 0 = 0$   
 $l++$

$v = 0 \quad l = 3$

$0 < FC[3] \quad \text{yes} \Rightarrow v = 2 \cdot 0 + 1 = 1$   
 $l++$

$v = 1 \quad l = 4$

$1 < FC[4] \quad \text{no} \Rightarrow \text{return symbol}[4; 1 - 0]$   
 $=> \textcircled{B}$

→ → →

$v = 0 \quad l = 1$

$0 < FC[1] \quad \text{yes} \Rightarrow v = 2 \cdot 0 + 1 = 1$   
 $l++;$

$v = 1 \quad l = 2$

$1 < FC[2] \quad \text{yes} \Rightarrow v = 2 \cdot 1 + 1 = 3$   
 $l++$

$v = 3 \quad l = 3$

$3 < FC[3] \quad \text{no} \Rightarrow \text{return symbol}[3; 3 - 1] = s[3, 2]$   
 $=> \textcircled{E}$

$v = 0 \quad l = 1$

$0 < FC[1] \quad \text{yes} \Rightarrow v = 2 \cdot 0 + 1 = 1 \quad l++$   
 $=> \textcircled{F}$

$v = 1 \quad l = 2$

$1 < FC[2] \quad \text{yes} \Rightarrow v = 2 \cdot 1 + 0 = 2 \quad l++$

$v = 2 \quad l = 3$

$2 < FC[3] \quad \text{no} \Rightarrow \text{return symbol}[3; 2 - 1] \Rightarrow \textcircled{D}$

$l$	1	2	3	4
$FC$	2	2	1	0

Symbol	0	1	2
1	=		
2	F	G	
3	C	D	E
4	A	B	

# ARITHMETIC CODING

- given a number  $x \in [0, 1)$

$\Rightarrow$  • positional representation:

$$\begin{array}{ccccccc} & b_1 & b_2 & b_3 & \dots & b_k & \dots \\ \downarrow & \downarrow & \downarrow & & & \downarrow & \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & & \frac{1}{2^k} & & \\ \text{"} & \text{"} & \text{"} & & \text{"} & & \\ 2^{-1} & 2^{-2} & 2^{-3} & & 2^{-k} & & \end{array}$$

(possible  $\infty$  representations)

$$b_i \in \{0, 1\}$$

$$\text{if representation is: } .101 = \frac{1}{2} + 0 + \frac{1}{8} = \frac{5}{8} \Rightarrow \text{DYADIC FRACTION}$$

$$\frac{\text{VALUE}}{2^k} \checkmark$$

COME BACK?

$$\frac{5}{8} = 2^3$$

- ▷ binary representation of  $\checkmark$
- $\underbrace{0 \dots 0}_{k} \text{ bin}(\checkmark)$

represent 5 in 3 bits  
• 101

$$\frac{5}{16} = \frac{5}{2^4} \Rightarrow 5 \text{ in 4 bits}$$

$$\begin{array}{c} .0101 \\ \downarrow \quad \downarrow \\ 1/4 \quad 1/16 \end{array} \quad \frac{1}{4} + \frac{1}{16} = \frac{5}{16} = \frac{45}{16}$$

- if the number is fractional, the dyadic fraction & binary representation:

CONVERTER ( $x, k$ )  $x \in [0, 1)$

number precision/accuracy (how many bits I want to use)

- consider  $2x$

if  $2x < 1 \Rightarrow$  output 0; repeat; (consider twice the value you get)  
 if  $2x \geq 1 \Rightarrow$  output 1;  $2x - 1$ ; repeat;

$$x = \frac{1}{3}$$

$$2x = \frac{2}{3} < 1 \Rightarrow \text{output 0; repeat}$$

$$2 \cdot \frac{2}{3} = \frac{4}{3} \geq 1 \Rightarrow \text{output 1; } 2x - 1 = \frac{4}{3} - 1 = \frac{1}{3}$$

$\Rightarrow$  representation is  $.0\overline{1}$

$\infty$  precn

e.g. encode  $\frac{5}{8}$

.101

$$2x = \frac{2 \otimes 1}{8} < 1 \Rightarrow \text{no output } 1$$

$$2 \cdot \frac{1}{4} = \frac{2}{4} > \frac{1}{2} < 1 \text{ yes output } 0 = \text{input}$$

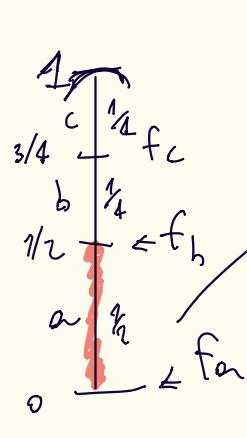
$$2 \cdot \frac{1}{2} = 1 < 1 \Rightarrow \text{output } 1$$

LEMMA 1: converter  $(x, k)$  makes an error  $2^{-k}$   $1-1=0$

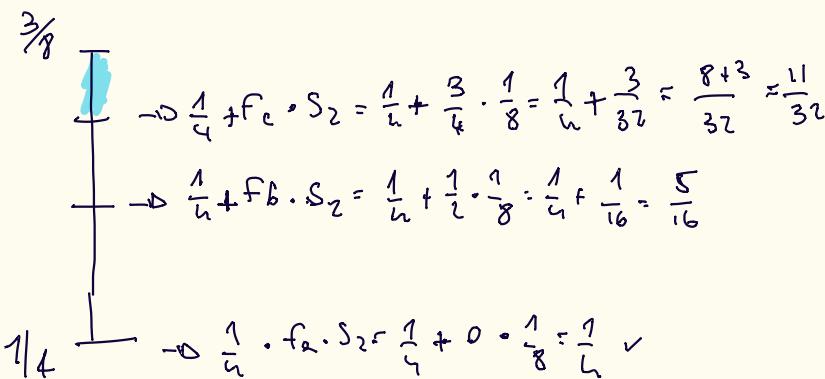
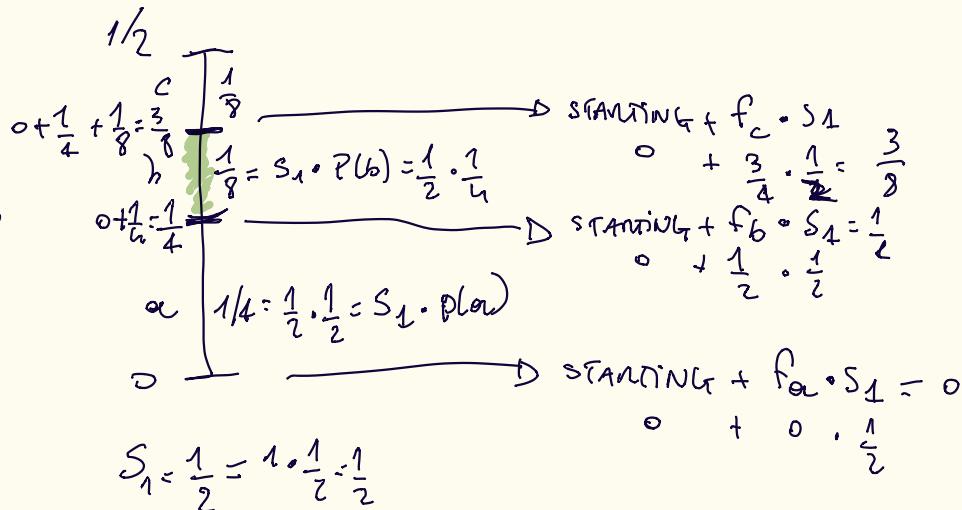
- given an alphabet of symbols with frequency

$$P(a) = \frac{1}{2}, P(b) = P(c) = \frac{1}{4} \Rightarrow f_a = 0, f_b = \frac{1}{2}, f_c = \frac{3}{4}$$

$T = a \ b \ c \ a$   $S_m = S_{m-1} \cdot P(\alpha)$  considered cumulative frequency of previous symbols  
 size of current interval  
 PARTITION ACCORDING  $P(a), P(b), P(c)$



$$S_0 = 1$$



$$S_2 = \frac{1}{2} \cdot P(b) = \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{8}$$

$$\begin{array}{l}
 \frac{3}{8} \\
 \text{---} \\
 \left| \begin{array}{l} \rightarrow \frac{11}{32} + f_C \cdot S_3 = \frac{11}{32} + \frac{3}{4} \cdot \frac{1}{32} = \frac{11}{32} + \frac{3}{128} = \frac{44+3}{128} = \frac{47}{128} \\ \rightarrow \frac{11}{32} + f_b \cdot S_3 = \frac{11}{32} + \frac{1}{2} \cdot \frac{1}{32} = \frac{11}{32} + \frac{1}{64} = \frac{22+1}{64} = \frac{23}{64} \end{array} \right. \\
 \frac{11}{32} \quad \rightarrow \frac{11}{32} + f_a \cdot S_3 = \frac{11}{32} + 0 \cdot \frac{1}{32} = \frac{11}{32}
 \end{array}$$

$$S_3 = S_2 \cdot p(c) = \frac{1}{8} \cdot \frac{1}{4} = \frac{1}{32}$$

$$\begin{array}{l}
 \frac{23}{64} \\
 \text{---} \\
 \left| \begin{array}{l} \rightarrow \frac{11}{32} + f_C \cdot S_4 = \frac{11}{32} + \frac{3}{4} \cdot \frac{11}{64} = \frac{11}{32} + \frac{33}{256} = \frac{44+33}{256} = \frac{88}{256} \\ \rightarrow \frac{11}{32} + f_b \cdot S_4 = \frac{11}{32} + \frac{1}{2} \cdot \frac{11}{64} = \frac{11}{32} + \frac{11}{128} = \frac{33+11}{128} = \frac{44}{128} \end{array} \right. \\
 \frac{11}{32} \quad \rightarrow \frac{11}{32} + f_a \cdot S_4 = \frac{11}{32} \\
 \frac{11}{64}
 \end{array}$$

$$S_4 = S_3 \cdot p(a) = \frac{11}{32} \cdot \frac{1}{2} = \frac{11}{64}$$

. ARITHMETIC CODE BUNDLES 2 integers  $\Rightarrow$  left end-point of the interval &  $l_m + \text{size}$

$$\circ \boxed{[l_m, l_m + s_m]}$$

$$\text{beginning: } l_0 = 0, S_0 = 1$$

in every STEP:

← correction in pos.  $n \Rightarrow$  to be encoded

$$\begin{aligned}
 \circ & \left\{ \begin{array}{l} S_{m+1} = S_m \cdot p(T[m+1]) \Rightarrow S_{m+1} = p(+[m+1]) \cdot S_m = \\ l_{m+1} = l_m + f_{T[m+1]} \cdot S_m = p(T[m+1]) \cdot p(T[m]) \cdot S_{m-1} = \\ = p(T[m+1]) \cdot p(T[m]) \cdot p(T[m-1]) \dots \cdot p(T[1]) \end{array} \right.
 \end{aligned}$$

$\Rightarrow$  since I'm looking for  $T = a \ b \ c \ a$

$$\Rightarrow \text{final size} \Rightarrow \frac{1}{2} \cdot \frac{1}{n} \cdot \frac{1}{n} \cdot \frac{1}{2} =$$

If  $T = \{a, b\}$

$$\rightarrow \left[ \frac{5}{16}, \frac{11}{32} \right] \quad \text{size of the interval}$$
$$\left[ \frac{5}{16}, \frac{5}{16} + \frac{1}{8} \cdot \frac{1}{5} \right]$$

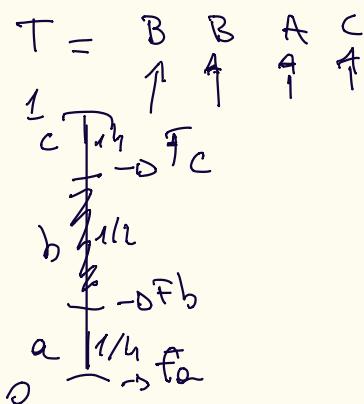
$\downarrow$   
 $\text{L}_2 \quad \text{B}_2 \quad \text{R}_2$

Arithmetic coding sends the middle element

$$\left[ \begin{array}{c} \text{L}_m + \text{S}_m \\ \text{B}_m \\ \text{R}_m \end{array} \right] \quad \text{L}_m + \frac{\text{S}_m}{2}$$

ENCODE WITH ARITHMETIC CODING THE TEXT  $T = B B A C$

(prob are not given)



$$P(a) = \frac{1}{4} \quad P(b) = \frac{2}{4} = \frac{1}{2} \quad P(c) = \frac{1}{4}$$

$$\underline{F_a = 0} \quad \underline{F_b = F_a + P(a)} = 0 + \frac{1}{4} = \boxed{\frac{1}{4}}$$

$$\underline{F_c = F_b + P(b)} = \frac{1}{2} + \frac{1}{4} = \boxed{\frac{3}{4}}$$

$$F_a = 0 \quad F_b = \frac{1}{4} \quad F_c = \frac{3}{4}$$

$$S_0 = 1 \quad \boxed{\frac{3}{4}}$$

$$\xrightarrow{\text{starting}} + F_c \cdot S_1 = \frac{1}{4} + \frac{3}{4} \cdot \frac{1}{2} = \frac{1}{4} + \frac{3}{8} = \frac{5}{8}$$

$$\xrightarrow{\text{starting}} + F_b \cdot S_1 = \frac{1}{4} + \frac{1}{4} \cdot \frac{1}{2} = \frac{1}{4} + \frac{1}{8} = \frac{3}{8}$$

$$\xrightarrow{\text{starting}} + F_a \cdot S_1 = \frac{1}{4} + 0 \cdot \frac{1}{2} = \frac{1}{4}$$

$$\begin{array}{r} 16 \\ 4 \\ \hline 64 \end{array}$$

$$S_1 = S_0 \cdot P(b) = 1 \cdot \frac{1}{2} = \frac{1}{2}$$

$$\boxed{\frac{1}{16}}$$

$$\xrightarrow{\text{starting}} + \frac{3}{8} + \frac{3}{4} \cdot \frac{1}{4} = \frac{3}{8} + \frac{3}{16} = \frac{9}{16}$$

$$\xrightarrow{\text{starting}} + \frac{3}{8} + \frac{3}{4} \cdot \frac{1}{16} = \frac{3}{8} + \frac{3}{64} = \frac{27}{64}$$

$$\xrightarrow{\text{starting}} + \frac{3}{8} + \frac{1}{4} \cdot \frac{1}{4} = \frac{3}{8} + \frac{1}{16} = \frac{7}{16}$$

$$\xrightarrow{\text{starting}} + \frac{3}{8} + \frac{1}{4} \cdot \frac{1}{16} = \frac{3}{8} + \frac{1}{64} = \frac{25}{64}$$

$$\xrightarrow{\text{starting}} + 0 \cdot \frac{1}{4} = \frac{3}{8}$$

$$\xrightarrow{\text{starting}} + 0 \cdot \frac{1}{16} = \frac{3}{8}$$

$$S_2 = S_1 \cdot P(b) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$$

$$S_3 = S_2 \cdot P(b) = \frac{1}{4} \cdot \frac{1}{2} = \frac{1}{16}$$

NON VA FATTO,  
farci

$$\xrightarrow{\text{starting}} + \frac{27}{64} + \frac{3}{4} \cdot \frac{1}{64} = \frac{27}{64} + \frac{3}{256} = \frac{108 + 3}{256} = \frac{111}{256}$$

$$\begin{array}{r} 64 \\ 4 \\ \hline 256 \end{array}$$

$|T|$   
STEPS

$$\xrightarrow{\text{starting}} + \frac{27}{64} + \frac{1}{4} \cdot \frac{1}{64} = \frac{27}{64} + \frac{1}{256} = \frac{109}{256}$$

$$\begin{array}{r} 256 \\ 4 \\ \hline 108 \end{array}$$

$$\xrightarrow{\text{starting}} + \frac{27}{64} + 0 \cdot \frac{1}{64} = \frac{27}{64}$$

$$S_4 = S_3 \cdot P(c) = \frac{1}{16} \cdot \frac{1}{4} = \frac{1}{64}$$

$$\begin{array}{c}
 \text{7/16} \\
 \text{C} \quad \overbrace{\frac{3}{8} + \frac{3}{4} \cdot \frac{1}{16}}^{\text{b}} = \frac{3}{8} + \frac{3}{64} = \frac{27}{64} \\
 \text{B} \quad \overbrace{\frac{3}{8} + \frac{1}{4} \cdot \frac{1}{16}}^{\text{e}} = \frac{3}{8} + \frac{1}{64} = \frac{25}{64} \\
 \text{E} \quad \overbrace{\frac{3}{8} + 0 \cdot \frac{1}{16}}^{\text{f}} = \frac{3}{8}
 \end{array}$$

$$S_2 := S_1 \cdot R_2 = \frac{1}{4} \cdot \frac{1}{4} = \frac{1}{16}$$

FINAL RANGE  
 $P(x) + \alpha$  (symbols)  
OUTPUT:  $x \in [l_{ij}; l_{i+1}j]$   
 $\rightarrow m = n$

- for instance choose middle element of range of C

$$\begin{aligned}
 &\Rightarrow \left( \frac{27}{64} + \frac{7}{16} \right) \cdot 2 = \\
 &= \left( \frac{27 + 28}{64} \right) \cdot 2 =
 \end{aligned}$$

$$\begin{aligned}
 &\Rightarrow \text{uncode } \frac{55}{128} \text{ in } \# \text{ bits } \lceil \log_2 \frac{27}{8} \rceil = \frac{55}{64} \cdot \frac{1}{2} = \frac{55}{128} \\
 &\text{converted } (x, \alpha) \qquad \qquad \qquad \lceil \log_2 \frac{27}{16} \rceil = \log_2 \\
 &\alpha = 2x
 \end{aligned}$$

if  $2x < 1 \Rightarrow$  OUTPUT 1 ; REPEAT;

else ( $2x \geq 1$ )  $\Rightarrow$  OUTPUT 0 ;  $\alpha - 1$  ; repeat

converted  $\left( \frac{55}{128} \right)$



• Theo: truncate to a # of bits  $d = \lceil \log_2 \frac{2}{s} \rceil$  bits

$$d = \lceil \log_2 \frac{2}{\frac{1}{64}} \rceil = \lceil \log_2 128 \rceil = 7 \text{ bits}$$

convention  $(\ell + \frac{s}{2}; d)$

convention  $(\frac{39}{128}; 7)$

$$\alpha = 2x$$

if  $(\alpha < 1)$  output 1; repeat

if  $(\alpha \geq 1)$  outputs;  $\alpha - 1$ ; repeat

$$\dots \Rightarrow 39 \text{ in } 7 \text{ bits} = 0.0100111$$

$\downarrow \quad \downarrow \downarrow \downarrow$

$$\frac{1}{5} \quad \frac{1}{32} \quad \frac{1}{64} \quad \frac{1}{128}$$

• Decode the compressed sequence  $<2, 1101>$  with arithmetic coding  
with  $p(a) = \frac{1}{2}$     $p(b) = \frac{1}{4}$     $p(c) = \frac{1}{4}$

$$\begin{array}{c} 1 \ 1 \ 0 \ 1 \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ \frac{1}{2} \quad \frac{1}{4} \quad \frac{1}{4} \quad \frac{1}{4} \\ \frac{1}{2} + \frac{1}{4} + \frac{1}{16} = \underbrace{\frac{8+4+1}{16}}_{\frac{13}{16}} \end{array} \Rightarrow \text{Search this}$$

start  $[0, 1)$

$$\begin{array}{c} 1 \\ \text{---} \\ C \\ \text{---} \\ 3 \\ \text{---} \\ 2 \\ \text{---} \\ 0 \\ \text{---} \\ S_0 = 1 \end{array} \rightarrow \bar{f}_a = 0$$

$$\begin{array}{c} 1 \\ \text{---} \\ C \\ \text{---} \\ 3 \\ \text{---} \\ 2 \\ \text{---} \\ e \\ \text{---} \\ 3/4 \\ \text{---} \\ s_1 = 1/4 \\ \text{---} \\ 3/16 \end{array} \rightarrow \begin{aligned} & \bar{f}_c = 3/4 \\ & \bar{f}_b = 1/2 \\ & \bar{f}_a = 0 \\ & \bar{f}_c = 3/4 \\ & \bar{f}_b = 3/4 + 1/4 = 3/4 + 1/16 = 15/16 \\ & \bar{f}_a = 3/4 + 1/2 \cdot 1/4 = 3/4 + 1/8 = 6/8 = 7/8 = 14/16 \end{aligned} \Rightarrow T = C, A$$

# ALGORITHMIC COMPLEXITY

$$P(a) = \frac{1}{8} = \frac{1}{2^3} \quad P(b) = \frac{1}{4} = \frac{1}{2^2} \quad P(c) = \frac{5}{8} = \frac{5}{2^3}$$

Specify which is the length in bits of  $T = aabbcc$

$$\log_2 \left\lceil \frac{2}{S} \right\rceil \rightarrow \text{last } \frac{\overbrace{up + down}^{\text{in round}}}{2}$$

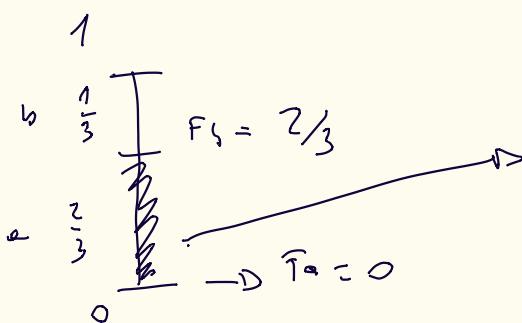
The length in bits is  $b = \lceil \log_2 \frac{2}{S} \rceil$  where  $S = \prod p[T(i)]$

$$\prod p[T(i)] = (p(a))^4 \cdot (p(b))^2 = \left(\frac{1}{8}\right)^4 \cdot \left(\frac{1}{4}\right)^2 = \\ = \left[\frac{1}{2^3}\right]^4 \cdot \left[\frac{1}{2^2}\right]^2 = \frac{1}{2^{12}} \cdot \frac{1}{2^4} = \frac{1}{2^{16}}$$

$$d = \lceil \log_2 \frac{2}{S} \rceil = \lceil \log_2 2 \cdot \frac{1}{2^{16}} \rceil = \lceil \log_2 \frac{1}{2^{16}} \rceil =$$

$$= \lceil \log_2 2 \cdot 2^{-16} \rceil = \lceil \log_2 2^{-15} \rceil = 17 \text{ bits}$$

Given  $T = \underbrace{AAB}_\text{arithmetic costly}$



$$P(a) = \frac{2}{3} \quad P(b) = \frac{1}{3}$$

$$\begin{aligned} & \text{STATE 1: } S_0 = 0 + \frac{2}{3} \cdot \frac{2}{3} = \frac{4}{9} \\ & \text{STATE 2: } S_1 = 0 + \frac{2}{3} \cdot \frac{2}{3} = 0 \end{aligned}$$

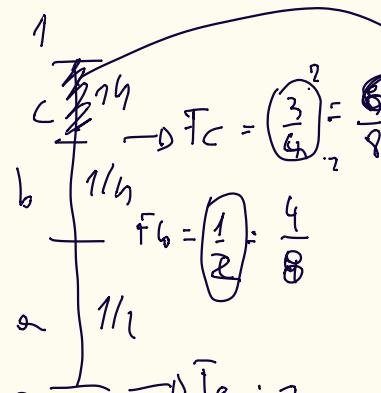
$$S_0 = 1 \cdot \frac{2}{3} : \frac{2}{3}$$

$$\begin{aligned} & \text{STATE 3: } S_2 = 0 + \frac{4}{9} \cdot \frac{2}{3} = \frac{8}{27} \\ & \text{STATE 4: } S_3 = 0 + \frac{8}{27} \cdot \frac{2}{3} = \frac{16}{81} \\ & \text{STATE 5: } S_4 = 0 + \frac{16}{81} \cdot \frac{2}{3} = \frac{32}{243} \\ & \text{STATE 6: } S_5 = 0 + \frac{32}{243} \cdot \frac{2}{3} = \frac{64}{729} \\ & \text{STATE 7: } S_6 = 0 + \frac{64}{729} \cdot \frac{2}{3} = \frac{128}{2187} \\ & \text{STATE 8: } S_7 = 0 + \frac{128}{2187} \cdot \frac{2}{3} = \frac{256}{6561} \\ & \text{STATE 9: } S_8 = 0 + \frac{256}{6561} \cdot \frac{2}{3} = \frac{512}{19683} \\ & \text{STATE 10: } S_9 = 0 + \frac{512}{19683} \cdot \frac{2}{3} = \frac{1024}{59049} \\ & \text{STATE 11: } S_{10} = 0 + \frac{1024}{59049} \cdot \frac{2}{3} = \frac{2048}{177147} \\ & \text{STATE 12: } S_{11} = 0 + \frac{2048}{177147} \cdot \frac{2}{3} = \frac{4096}{531471} \\ & \text{STATE 13: } S_{12} = 0 + \frac{4096}{531471} \cdot \frac{2}{3} = \frac{8192}{1594413} \\ & \text{STATE 14: } S_{13} = 0 + \frac{8192}{1594413} \cdot \frac{2}{3} = \frac{16384}{4783239} \\ & \text{STATE 15: } S_{14} = 0 + \frac{16384}{4783239} \cdot \frac{2}{3} = \frac{32768}{14349717} \\ & \text{STATE 16: } S_{15} = 0 + \frac{32768}{14349717} \cdot \frac{2}{3} = \frac{65536}{43049151} \\ & \text{STATE 17: } S_{16} = 0 + \frac{65536}{43049151} \cdot \frac{2}{3} = \frac{131072}{129147453} \\ & \text{STATE 18: } S_{17} = 0 + \frac{131072}{129147453} \cdot \frac{2}{3} = \frac{262144}{387442359} \\ & \text{STATE 19: } S_{18} = 0 + \frac{262144}{387442359} \cdot \frac{2}{3} = \frac{524288}{1162327077} \\ & \text{STATE 20: } S_{19} = 0 + \frac{524288}{1162327077} \cdot \frac{2}{3} = \frac{1048576}{3487081231} \\ & \text{STATE 21: } S_{20} = 0 + \frac{1048576}{3487081231} \cdot \frac{2}{3} = \frac{2097152}{10461243793} \\ & \text{STATE 22: } S_{21} = 0 + \frac{2097152}{10461243793} \cdot \frac{2}{3} = \frac{4194304}{31383721381} \\ & \text{STATE 23: } S_{22} = 0 + \frac{4194304}{31383721381} \cdot \frac{2}{3} = \frac{8388608}{94151144143} \\ & \text{STATE 24: } S_{23} = 0 + \frac{8388608}{94151144143} \cdot \frac{2}{3} = \frac{16777216}{288453432429} \\ & \text{STATE 25: } S_{24} = 0 + \frac{16777216}{288453432429} \cdot \frac{2}{3} = \frac{33554432}{865359864887} \\ & \text{STATE 26: } S_{25} = 0 + \frac{33554432}{865359864887} \cdot \frac{2}{3} = \frac{67108864}{2595819594661} \\ & \text{STATE 27: } S_{26} = 0 + \frac{67108864}{2595819594661} \cdot \frac{2}{3} = \frac{134217728}{7787458789323} \\ & \text{STATE 28: } S_{27} = 0 + \frac{134217728}{7787458789323} \cdot \frac{2}{3} = \frac{268435456}{23362376367969} \\ & \text{STATE 29: } S_{28} = 0 + \frac{268435456}{23362376367969} \cdot \frac{2}{3} = \frac{536870912}{70087152103907} \\ & \text{STATE 30: } S_{29} = 0 + \frac{536870912}{70087152103907} \cdot \frac{2}{3} = \frac{1073741824}{210261456311721} \\ & \text{STATE 31: } S_{30} = 0 + \frac{1073741824}{210261456311721} \cdot \frac{2}{3} = \frac{2147483648}{630522912635163} \\ & \text{STATE 32: } S_{31} = 0 + \frac{2147483648}{630522912635163} \cdot \frac{2}{3} = \frac{4294967296}{189154583790549} \\ & \text{STATE 33: } S_{32} = 0 + \frac{4294967296}{189154583790549} \cdot \frac{2}{3} = \frac{8589934592}{567463751371647} \\ & \text{STATE 34: } S_{33} = 0 + \frac{8589934592}{567463751371647} \cdot \frac{2}{3} = \frac{17179869184}{1702386254115341} \\ & \text{STATE 35: } S_{34} = 0 + \frac{17179869184}{1702386254115341} \cdot \frac{2}{3} = \frac{34359738368}{5107158762345023} \\ & \text{STATE 36: } S_{35} = 0 + \frac{34359738368}{5107158762345023} \cdot \frac{2}{3} = \frac{68719476736}{1532147628703507} \\ & \text{STATE 37: } S_{36} = 0 + \frac{68719476736}{1532147628703507} \cdot \frac{2}{3} = \frac{137438953472}{4596492856107514} \\ & \text{STATE 38: } S_{37} = 0 + \frac{137438953472}{4596492856107514} \cdot \frac{2}{3} = \frac{274877906944}{1378898551832258} \\ & \text{STATE 39: } S_{38} = 0 + \frac{274877906944}{1378898551832258} \cdot \frac{2}{3} = \frac{549755813888}{4157797105496776} \\ & \text{STATE 40: } S_{39} = 0 + \frac{549755813888}{4157797105496776} \cdot \frac{2}{3} = \frac{1099511627776}{1247339421649032} \\ & \text{STATE 41: } S_{40} = 0 + \frac{1099511627776}{1247339421649032} \cdot \frac{2}{3} = \frac{2199023255552}{3744678845947064} \\ & \text{STATE 42: } S_{41} = 0 + \frac{2199023255552}{3744678845947064} \cdot \frac{2}{3} = \frac{4398046511104}{11234036137841216} \\ & \text{STATE 43: } S_{42} = 0 + \frac{4398046511104}{11234036137841216} \cdot \frac{2}{3} = \frac{8796093022208}{34102072413523648} \\ & \text{STATE 44: } S_{43} = 0 + \frac{8796093022208}{34102072413523648} \cdot \frac{2}{3} = \frac{17592186044416}{10230621474507096} \\ & \text{STATE 45: } S_{44} = 0 + \frac{17592186044416}{10230621474507096} \cdot \frac{2}{3} = \frac{35184372088832}{3070184392352128} \\ & \text{STATE 46: } S_{45} = 0 + \frac{35184372088832}{3070184392352128} \cdot \frac{2}{3} = \frac{70368744177664}{921055817450704} \\ & \text{STATE 47: } S_{46} = 0 + \frac{70368744177664}{921055817450704} \cdot \frac{2}{3} = \frac{140737488355328}{276316745235212} \\ & \text{STATE 48: } S_{47} = 0 + \frac{140737488355328}{276316745235212} \cdot \frac{2}{3} = \frac{281474976710656}{828633490405636} \\ & \text{STATE 49: } S_{48} = 0 + \frac{281474976710656}{828633490405636} \cdot \frac{2}{3} = \frac{562949953421312}{2485900471216908} \\ & \text{STATE 50: } S_{49} = 0 + \frac{562949953421312}{2485900471216908} \cdot \frac{2}{3} = \frac{1125899906842624}{7457700943653716} \\ & \text{STATE 51: } S_{50} = 0 + \frac{1125899906842624}{7457700943653716} \cdot \frac{2}{3} = \frac{2251799813685248}{22373402821951452} \\ & \text{STATE 52: } S_{51} = 0 + \frac{2251799813685248}{22373402821951452} \cdot \frac{2}{3} = \frac{4503599627370496}{67120805445854304} \\ & \text{STATE 53: } S_{52} = 0 + \frac{4503599627370496}{67120805445854304} \cdot \frac{2}{3} = \frac{9007199254740992}{201362416337562912} \\ & \text{STATE 54: } S_{53} = 0 + \frac{9007199254740992}{201362416337562912} \cdot \frac{2}{3} = \frac{18014398509481984}{603724848612688736} \\ & \text{STATE 55: } S_{54} = 0 + \frac{18014398509481984}{603724848612688736} \cdot \frac{2}{3} = \frac{36028797018963968}{1811449695838066208} \\ & \text{STATE 56: } S_{55} = 0 + \frac{36028797018963968}{1811449695838066208} \cdot \frac{2}{3} = \frac{72057594037927936}{543434938946419864} \\ & \text{STATE 57: } S_{56} = 0 + \frac{72057594037927936}{543434938946419864} \cdot \frac{2}{3} = \frac{144115188075855872}{162986976683925952} \\ & \text{STATE 58: } S_{57} = 0 + \frac{144115188075855872}{162986976683925952} \cdot \frac{2}{3} = \frac{288230376151711744}{488973953347757888} \\ & \text{STATE 59: } S_{58} = 0 + \frac{288230376151711744}{488973953347757888} \cdot \frac{2}{3} = \frac{576460752303423488}{1466911906643273664} \\ & \text{STATE 60: } S_{59} = 0 + \frac{576460752303423488}{1466911906643273664} \cdot \frac{2}{3} = \frac{1152921504606846976}{4393835719929810992} \\ & \text{STATE 61: } S_{60} = 0 + \frac{1152921504606846976}{4393835719929810992} \cdot \frac{2}{3} = \frac{2305843009213693952}{1318747143309940332} \\ & \text{STATE 62: } S_{61} = 0 + \frac{2305843009213693952}{1318747143309940332} \cdot \frac{2}{3} = \frac{4611686018427387904}{4037294476929801064} \\ & \text{STATE 63: } S_{62} = 0 + \frac{4611686018427387904}{4037294476929801064} \cdot \frac{2}{3} = \frac{9223372036854775808}{12074588920789403192} \\ & \text{STATE 64: } S_{63} = 0 + \frac{9223372036854775808}{12074588920789403192} \cdot \frac{2}{3} = \frac{18446744073709551616}{36149176762368209576} \\ & \text{STATE 65: } S_{64} = 0 + \frac{18446744073709551616}{36149176762368209576} \cdot \frac{2}{3} = \frac{36893488147419103232}{10849755328710462872} \\ & \text{STATE 66: } S_{65} = 0 + \frac{36893488147419103232}{10849755328710462872} \cdot \frac{2}{3} = \frac{73786976294838206464}{32549267686134882616} \\ & \text{STATE 67: } S_{66} = 0 + \frac{73786976294838206464}{32549267686134882616} \cdot \frac{2}{3} = \frac{147573952589676412928}{97647835054449608048} \\ & \text{STATE 68: } S_{67} = 0 + \frac{147573952589676412928}{97647835054449608048} \cdot \frac{2}{3} = \frac{295147905179352825856}{295343670163348824144} \\ & \text{STATE 69: } S_{68} = 0 + \frac{295147905179352825856}{295343670163348824144} \cdot \frac{2}{3} = \frac{590295810358705651712}{885987340489995472432} \\ & \text{STATE 70: } S_{69} = 0 + \frac{590295810358705651712}{885987340489995472432} \cdot \frac{2}{3} = \frac{1180591620717411303424}{2657954681469986217296} \\ & \text{STATE 71: } S_{70} = 0 + \frac{1180591620717411303424}{2657954681469986217296} \cdot \frac{2}{3} = \frac{2361183241434822606848}{8173859364339952454592} \\ & \text{STATE 72: } S_{71} = 0 + \frac{2361183241434822606848}{8173859364339952454592} \cdot \frac{2}{3} = \frac{4722366482869645213696}{24519718092619857363784} \\ & \text{STATE 73: } S_{72} = 0 + \frac{4722366482869645213696}{24519718092619857363784} \cdot \frac{2}{3} = \frac{9444732965739290427392}{73559436187859514591568} \\ & \text{STATE 74: } S_{73} = 0 + \frac{9444732965739290427392}{73559436187859514591568} \cdot \frac{2}{3} = \frac{18889465931478580854784}{220678208563578543774704} \\ & \text{STATE 75: } S_{74} = 0 + \frac{18889465931478580854784}{220678208563578543774704} \cdot \frac{2}{3} = \frac{37778931862957161709568}{661356416686735081324144} \\ & \text{STATE 76: } S_{75} = 0 + \frac{37778931862957161709568}{661356416686735081324144} \cdot \frac{2}{3} = \frac{75557863725914323419136}{2002668050059105243972432} \\ & \text{STATE 77: } S_{76} = 0 + \frac{75557863725914323419136}{2002668050059105243972432} \cdot \frac{2}{3} = \frac{151115727451828646838272}{6007994150177315731917296} \\ & \text{STATE 78: } S_{77} = 0 + \frac{151115727451828646838272}{6007994150177315731917296} \cdot \frac{2}{3} = \frac{302231454903657293676544}{18023988300524635765751584} \\ & \text{STATE 79: } S_{78} = 0 + \frac{302231454903657293676544}{18023988300524635765751584} \cdot \frac{2}{3} = \frac{604462909807314587353088}{5404795690157387729225472} \\ & \text{STATE 80: } S_{79} = 0 + \frac{604462909807314587353088}{5404795690157387729225472} \cdot \frac{2}{3} = \frac{1208925819614629174706176}{16214386770471163187476448} \\ & \text{STATE 81: } S_{80} = 0 + \frac{1208925819614629174706176}{16214386770471163187476448} \cdot \frac{2}{3} = \frac{241785163922925834941232}{48643159211413489562429344} \\ & \text{STATE 82: } S_{81} = 0 + \frac{241785163922925834941232}{48643159211413489562429344} \cdot \frac{2}{3} = \frac{483570327845851669882464}{145929477434240978687287088} \\ & \text{STATE 83: } S_{82} = 0 + \frac{483570327845851669882464}{145929477434240978687287088} \cdot \frac{2}{3} = \frac{967140655691703339764928}{437788432302722935061861264} \\ & \text{STATE 84: } S_{83} = 0 + \frac{967140655691703339764928}{437788432302722935061861264} \cdot \frac{2}{3} = \frac{193428131138340667952956}{131336476767574311687253752} \\ & \text{STATE 85: } S_{84} = 0 + \frac{193428131138340667952956}{131336476767574311687253752} \cdot \frac{2}{3} = \frac{386856262276681335905912}{10006918141405945934940296} \\ & \text{STATE 86: } S_{85} = 0 + \frac{386856262276681335905912}{10006918141405945934940296} \cdot \frac{2}{3} = \frac{773712524553362671811824}{5003459070702972967470148} \\ & \text{STATE 87: } S_{86} = 0 + \frac{773712524553362671811824}{5003459070702972967470148} \cdot \frac{2}{3} = \frac{1547425049106725343623648}{2501729535351486483735074} \\ & \text{STATE 88: } S_{87} = 0 + \frac{1547425049106725343623648}{2501729535351486483735074} \cdot \frac{2}{3} = \frac{3094850098213450687247296}{12508647676757432418675372} \\ & \text{STATE 89: } S_{88} = 0 + \frac{3094850098213450687247296}{12508647676757432418675372} \cdot \frac{2}{3} = \frac{6189700196426901374494592}{6254323838378716209337686} \\ & \text{STATE 90: } S_{89} = 0 + \frac{6189700196426901374494592}{6254323838378716209337686} \cdot \frac{2}{3} = \frac{12379400392853802748989184}{3127161919189358104668843} \\ & \text{STATE 91: } S_{90} = 0 + \frac{12379400392853802748989184}{3127161919189358104668843} \cdot \frac{2}{3} = \frac{24758800785707605497978368}{15635809595946790523344215} \\ & \text{STATE 92: } S_{91} = 0 + \frac{24758800785707605497978368}{15635809595946790523344215} \cdot \frac{2}{3} = \frac{49517601571415210995956736}{78179047979733952616721075} \\ & \text{STATE 93: } S_{92} = 0 + \frac{49517601571415210995956736}{78179047979733952616721075} \cdot \frac{2}{3} = \frac{99035203142830421991913472}{390895239898669763083605375} \\ & \text{STATE 94: } S_{93} = 0 + \frac{99035203142830421991913472}{390895239898669763083605375} \cdot \frac{2}{3} = \frac{198070406285660843983826944}{1954476199493348815418027875} \\ & \text{STATE 95: } S_{94} = 0 + \frac{198070406285660843983826944}{1954476199493348815418027875} \cdot \frac{2}{3} = \frac{396140812571321687967653888}{9772380997466744077090139375} \\ & \text{STATE 96: } S_{95} = 0 + \frac{396140812571321687967653888}{9772380997466744077090139375} \cdot \frac{2}{3} = \frac{792281625142643375935307776}{48861904987333720385450696875} \\ & \text{STATE 97: } S_{96} = 0 + \frac{792281625142643375935307776}{48861904987333720385450696875} \cdot \frac{2}{3} = \frac{1584563250285286751870615552}{244309524936668601927253484375} \\ & \text{STATE 98: } S_{97} = 0 + \frac{1584563250285286751870615552}{244309524936668601927253484375} \cdot \frac{2}{3} = \frac{3169126500570573503741231104}{1221547624683343009636267421875} \\ & \text{STATE 99: } S_{98} = 0 + \frac{3169126500570573503741231104}{1221547624683343009636267421875} \cdot \frac{2}{3} = \frac{6338253001141147007482462208}{6107738123416715048181337109375} \\ & \text{STATE 100: } S_{99} = 0 + \frac{6338253001141147007482462208}{6107738123416715048181337109375} \cdot \frac{2}{3} = \frac{12676506002282294014964924416}{30538690617083575240906685546875} \\ & \text{STATE 101: } S_{100} = 0 + \frac{12676506002282294014964924416}{30538690617083575240906685546875} \cdot \frac{2}{3} = \frac{25353012004564588029929848832}{152693453085417875204533427734375} \\ & \text{STATE 102: } S_{101} = 0 + \frac{253530120045645$$



Arithmetical coding, decompress

$$P(a) = \frac{1}{2} \quad P(b) = \frac{1}{4} \quad P(c) = \frac{1}{4}$$



$$S_0: 1$$

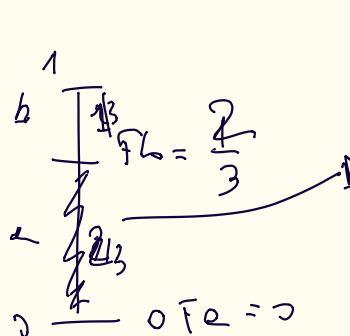
$$\begin{aligned} & \text{Root: } 1 \\ & \text{Left branch: } a \rightarrow T_a = 0 \\ & \text{Right branch: } b \rightarrow T_b = 1 \\ & \text{Left branch of } b: 0 \rightarrow T_{b0} = \frac{1}{4} \\ & \text{Right branch of } b: 1 \rightarrow T_{b1} = \frac{3}{4} \\ & \text{Left branch of } T_{b1}: 0 \rightarrow T_{b10} = \frac{3}{16} \\ & \text{Right branch of } T_{b1}: 1 \rightarrow T_{b11} = \frac{1}{16} \\ & S_1 = 1 \cdot P(a) = \frac{1}{2} \end{aligned}$$

$$\begin{aligned} & \text{Root: } 1 \\ & \text{Left branch: } \frac{1}{2} \\ & \text{Right branch: } \frac{1}{4} \\ & \text{Left branch of } \frac{1}{4}: \frac{1}{8} \\ & \text{Right branch of } \frac{1}{4}: \frac{1}{16} \\ & \frac{1}{2} + \frac{1}{4} + \frac{1}{8} \\ & = \frac{7}{8} \end{aligned}$$

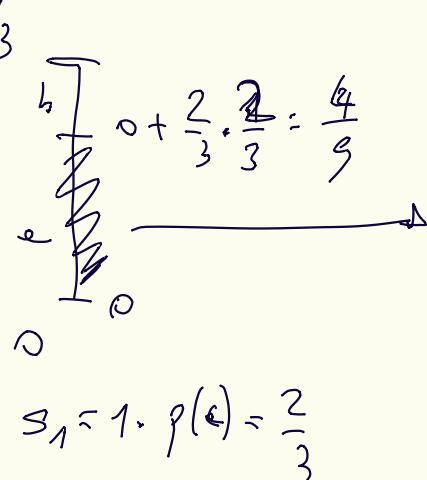
$$T = C \quad B$$

# ARITHMETIC CODING

$$T = \underline{a} \underline{e} \underline{b} \quad p(a) = \frac{2}{3} \quad p(b) = \frac{1}{3}$$



$$S_0 = 1$$



encode

$$b = \left( \log_2 \frac{2}{S_3} \right)$$

$$S_3 = \frac{4}{9} \cdot \frac{1}{3} = \frac{4}{27}$$

$$b = \left[ \log_2 2 \cdot \frac{27}{4} \right] = \left( \log_2 13, 5 \right) = 4 \text{ bits}$$

converter  $(\frac{10}{27}, 4)$

$$2 \cdot \frac{10}{27} = \frac{20}{27} < 1 \Rightarrow \text{output } 0$$

$$2 \cdot \frac{20}{27} = \frac{40}{27} > 1 \Rightarrow \text{output } 1 \quad \frac{40}{27} - 1 = \frac{13}{27}$$

$$\frac{13}{27} \cdot 2 = \frac{26}{27} < 1 \Rightarrow \text{output } 0$$

$\Rightarrow 0101, 3>$

$$\frac{26}{27} \cdot 2 = \frac{52}{27} > 1 \Rightarrow \text{output } 1$$