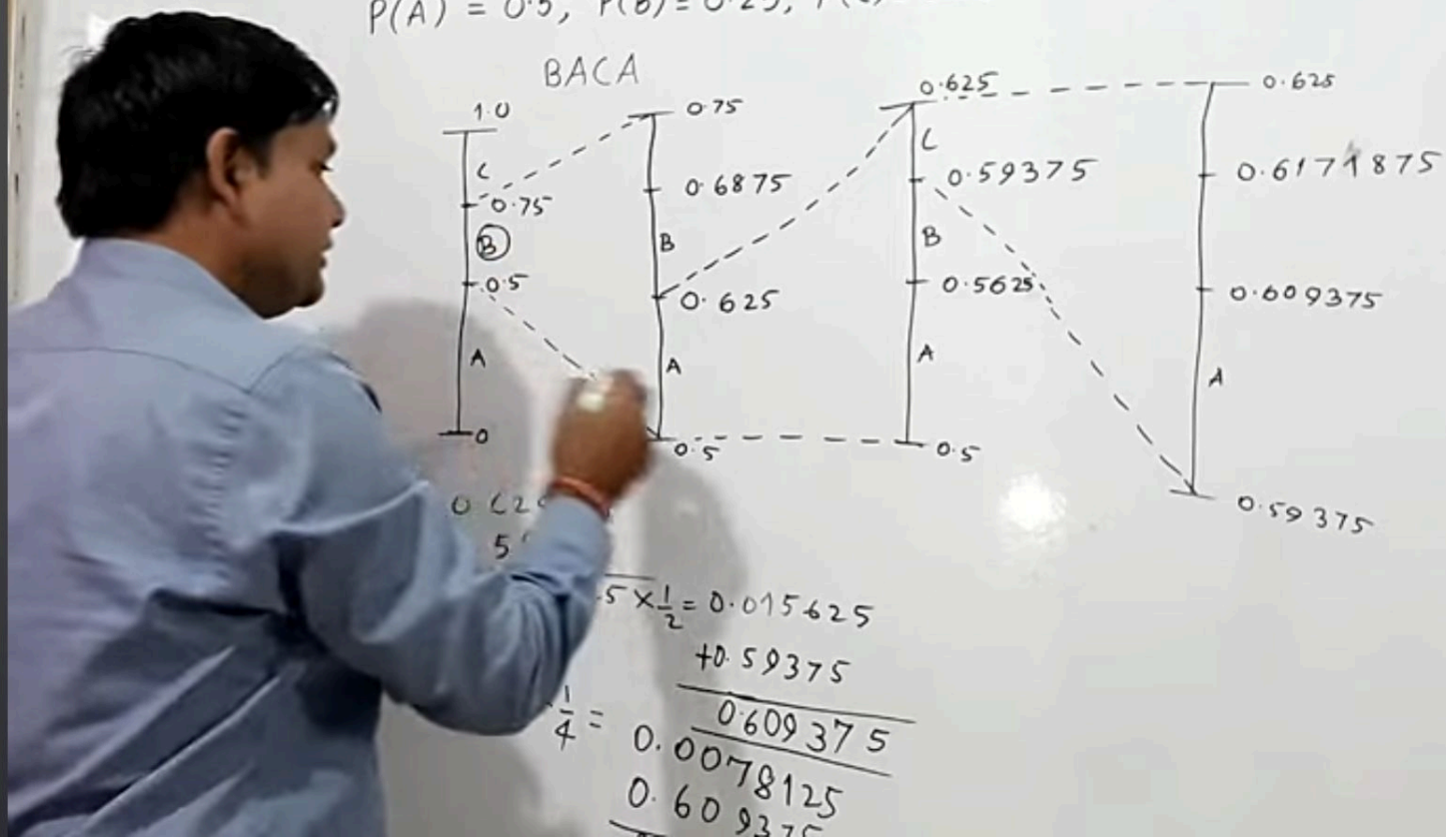
Arithmetic Coding

Arithmatic Coding

$P(A) = 0.5,\ P(B) = 0.25,\ P(C) = 0.25$



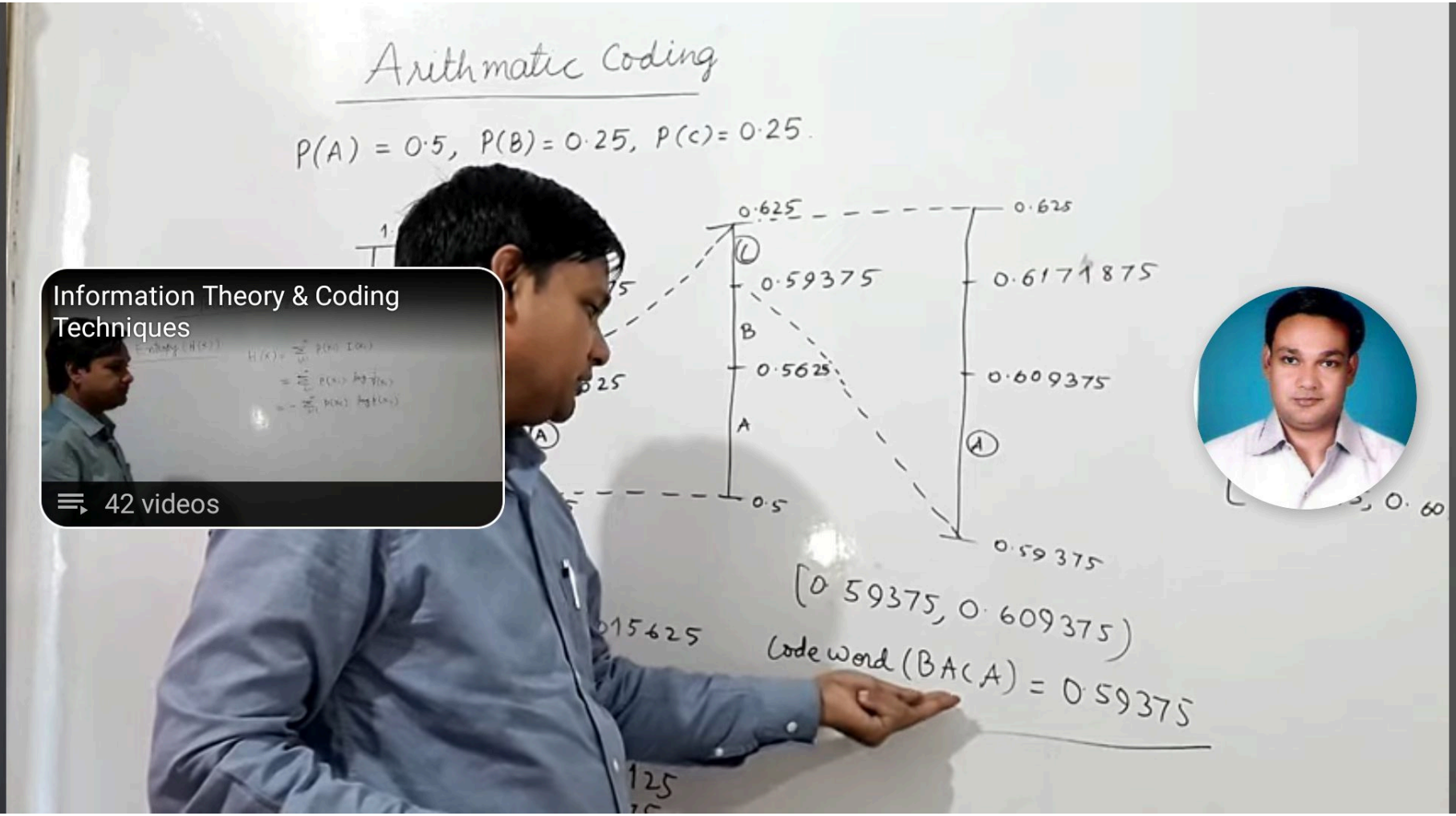
$\times \frac{1}{2} = 0.015625$

$+0.59375$

0.609375

$\frac{1}{4} = 0.0078125$

0.609375

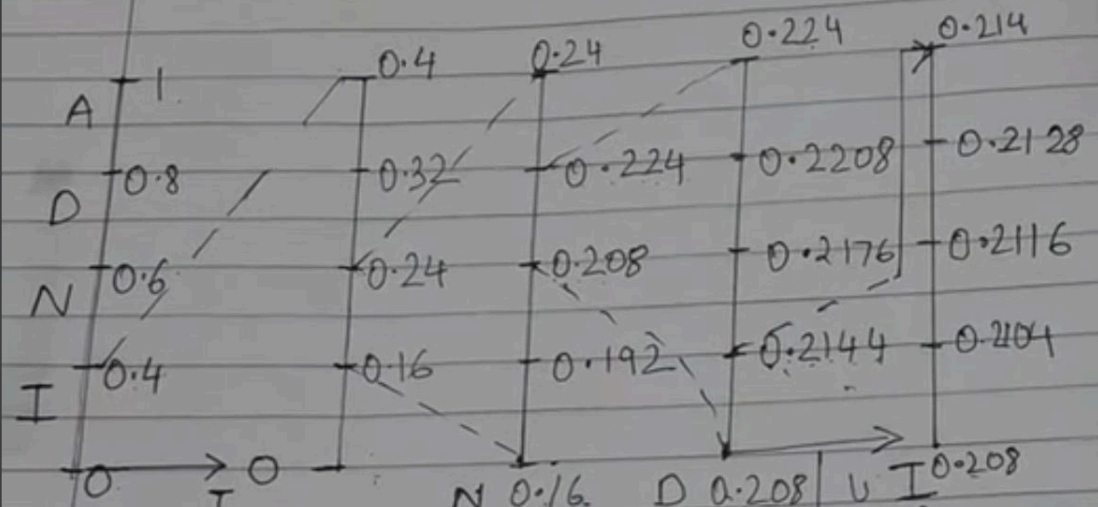

Arithmatic Coding
P(A) = 0·5, P(B) = 0·25, P(c) = 0·25
0·625
0·59375
B
0·5625
A
0·5
0·625
0·6171875
0·609375
0·59375
[0·59375, 0·609375)
Codeword (BA(A) = 0·59375
Information Theory & Coding Techniques
42 videos

ARITHMETIC CODING

message INDIA

Symbol	Prob	CDF
I	2/5 = 0.4	0.4
N	1/5 = 0.2	0.6
D	1/5 = 0.2	0.8
A	1/5 = 0.2	1

D	1/5 = 0.2	0.8
A	1/5 = 0.2	1



$UL = LL + d(P)(d)$

$= 0 + 0.4 \times 0.4$

$UL = LL + D \times P(d)$

$0 + 0.4 \times 0.6$

N $UL = LL + D \times P(d)$

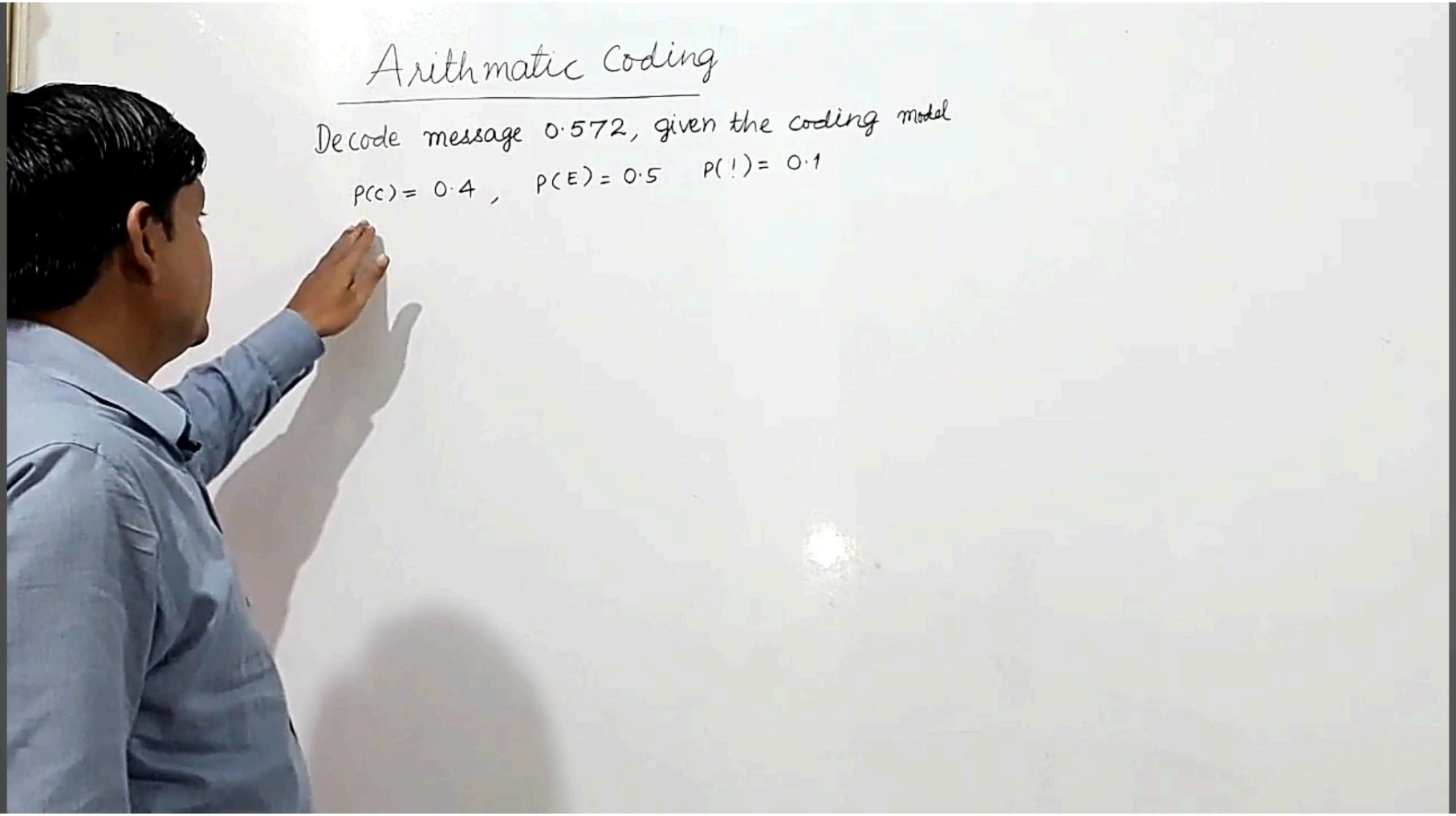
$UL = LL + d \times P(d)$

$= 0.208 + 0.016 \times 0.4$

$UL = LL + d \times P(d)$

$= .208 + 0.006 \times 0.4$

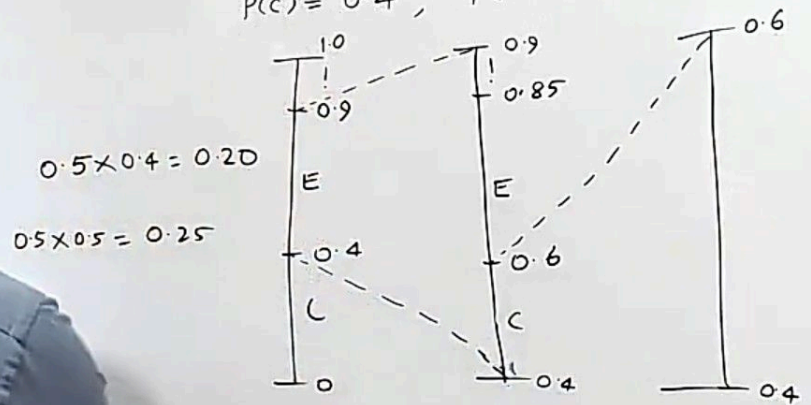
Arithmetic Decoding


Arithmatic Coding
Decode message 0.572, given the coding model
P(C) = 0.4 , P(E) = 0.5 P(!) = 0.1

Arithmatic Coding

Decode message 0.572, given the coding model

P(C) = 0.4 , P(E) = 0.5 P(!) = 0.1



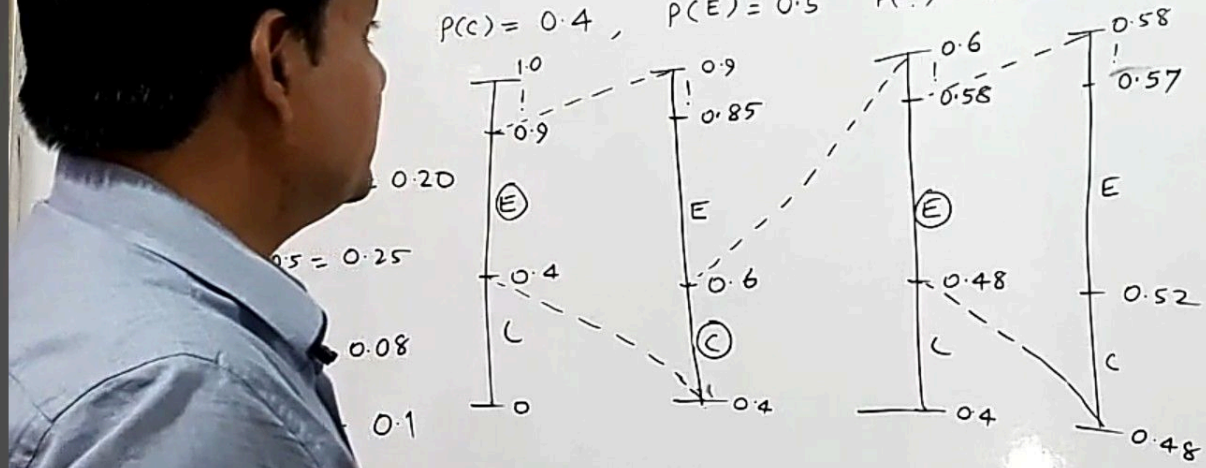
0.5×0.4 = 0.20

0.5×0.5 = 0.25

Arithmatic Coding

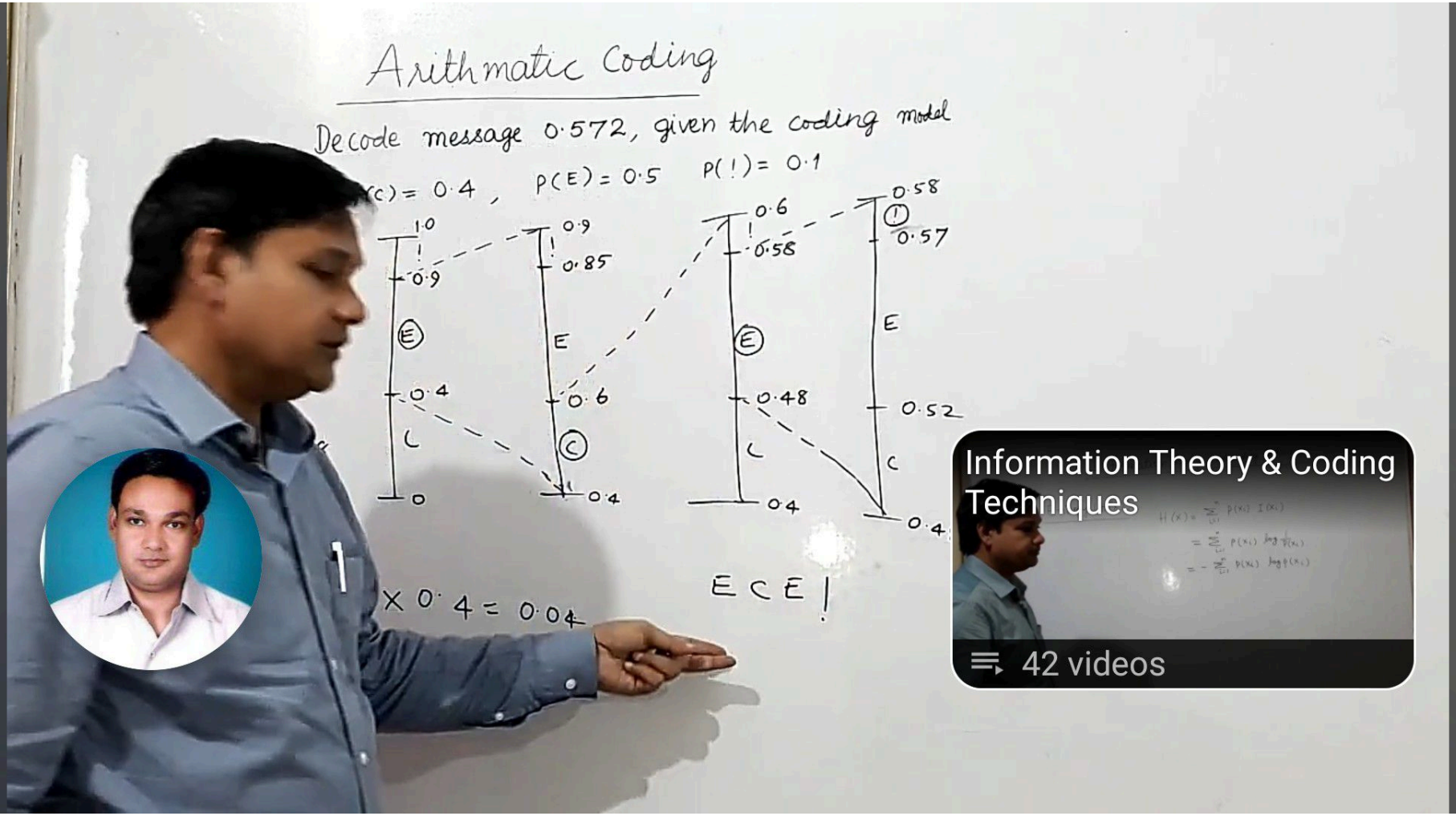
Decode message 0.572, given the coding model

P(C) = 0.4 , P(E) = 0.5 P(!) = 0.1



0.1 X 0.4 = 0.04

0.1 X 0.5 = 0.05


Arithmatic Coding
Decode message 0.572, given the coding model
P(C) = 0.4 , P(E) = 0.5 P(!) = 0.1
1.0
0.9
0
0.9
0.85
0.4
0.6
0.6
0.58
0.48
0.4
0.58
0.57
0.52
0.4
E
C
!
1
X 0.4 = 0.04
E C E !
Information Theory & Coding Techniques
42 videos

LZW Encoding

LZW

<i>

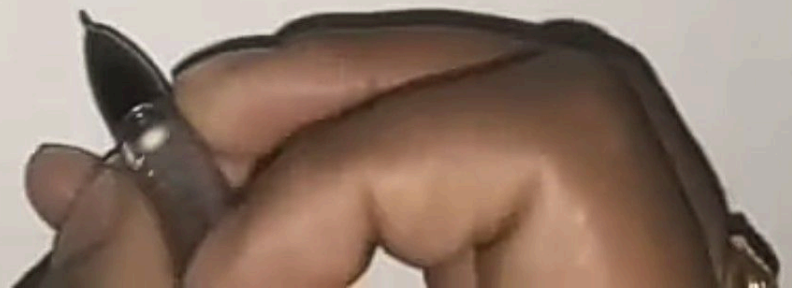
Q- Encode : wabbaβwabbaβwabbaβwabbaβw

Initial given dictionary :

Index	Entry
1	β
2	a
3	b
4	o
5	w

→

Index	Entry
1	β
2	a
3	b
4	o
5	w



LZW

<i>

Q- Encode : w a b b a ƀ w a b b a ƀ w a b b a ƀ w a b b a ƀ w
5

Initial given dictionary :

Index	Entry
1	ƀ
2	a
3	b
4	o
5	w

« 5,2

→

Index	Entry
1	ƀ
2	a
3	b
4	o
5	w
6	wa
7	ab

LZW

<i>

Q- Encode : w a bb a ß w a b b a ß w a bb a ß w a b b a ß w a bb a ß w
5

Initial given dictionary :

Index	Entry
1	ß
2	a
3	b
4	o
5	w

" 5,2,3,

→

Index	Entry
1	ß
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb

LZW

<i>

Q- Encode : w a b b a ƀ w a b b a ƀ w a b b a ƀ w a b b a ƀ w
(5)

Initial given dictionary :

Index	Entry
1	ƀ
2	a
3	b
4	o
5	w

" 5, 2, 3, 3, 2,

→

Index	Entry
1	ƀ
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb
9	ba
10	aƀ

LZW

<i>

Q- Encode : w a bb a ƀ wa bb a ƀ wa bb a ƀ wa bb a ƀ w
(5)

Initial given dictionary :

Index	Entry
1	ƀ
2	a
3	b
4	o
5	w

" 5, 2, 3, 3, 2, 1, 6, 8, 10, 12, 9, 11, 7, 16 "

→

Index	Entry
1	ƀ
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb
9	ba
10	aƀ
11	ƀw
12	wab
13	bba
14	aƀw
15	wabb
16	baƀ
17	ƀwa
18	abb
19	baƀw

LZW Encoding

Seq. - a b a b b a b c a b a b b a

Initial LZW dictionary

index	entry
1	a
2	b
3	c

Encoded o/p	Dictionary	
	index	entry
–	1	a
–	2	b
–	3	c
1	4	ab
2	5	ba
4	6	abb
5	7	bab
2	8	bc
3	9	ca
4	10	aba
6	11	abba
1	–	–

1 2 4 5 2 3 4 6 1 ans.

Best ✓

LZW Decoding

LZW decoding

Q- Sequence : " 3, 1, 4, 6, 8, 4, 2, 1, 2, 5, 10, 6, 11, 13, 6 "

Initial seq. dictionary :

Index	entry
1	a
2	ß
3	γ
4	t

→ Dictionary :

Index	Entry
1	a
2	ß
3	γ
4	t

LZW decoding

Q- Sequence : " 3, 1, 4, 6, 8, 4, 2, 1, 2, 5, 10, 6, 11, 13, 6 "

Initial seq. dictionary :

Index	entry
1	a
2	b
3	r
4	t

} " rat

→ Dictionary :

Index	Entry
1	a
2	b
3	r
4	t

LZW decoding

Q- Sequence : " 3, 1, 4, 6, 8, 4, 2, 1, 2, 5, 10, 6, 11, 13, 6 "

Initial seq. dictionary :

Index	entry
1	a
2	b
3	r
4	t

} " ratat

→ Dictionary :

Index	Entry
1	a
2	b
3	r
4	t
5	ra
6	at

LZW decoding

Q- Sequence : " 3, 1, 4, 6, 8, 4, 2, 1, 2, 5, 10, 6, 11, 13, 6 "

Initial seq. dictionary :

Index	entry
1	a
2	b
3	r
4	t

" ratat

→ Dictionary :

Index	Entry
1	a
2	b
3	r ✓
4	t
5	ra
6	at ✓
7	ta
8	

LZW decoding

Q- Sequence : " 3, 1, 4, 6, 8, 4, 2, 1, 2, 5, 10, 6, 11, 13, 6 "

Initial seq. dictionary :

Index	entry
1	a
2	b
3	r
4	t

} " ratat

→ Dictionary :

Index	Entry
1	a
2	b
3	r
4	t
5	ra
6	at
7	ta
8	at_

LZW decoding

Q- Sequence : " 3, 1, 4, 6, 8, 4, 2, 1, 2, 5, 10, 6, 11, 13, 6 "

Initial seq. dictionary :

Index	entry
1	a
2	b
3	r
4	t

" ratatat

→ Dictionary :

Index	Entry
1	a
2	b
3	r
4	t
5	ra
6	at
7	ta
8	at_

LZW decoding

Q- Sequence : " 3, 1, 4, 6, 8, 4, 2, 1, 2, 5, 10, 6, 11, 13, 6 "

Initial seq. dictionary :

Index	entry
1	a
2	b
3	r
4	t

} " ratatat

→ Dictionary :

Index	Entry
1	a
2	b
3	r
4	t
5	ra
6	at
7	ta
8	ata

LZW decoding

Q- Sequence : " 3, 1, 4, 6, 8, 4, 2, 1, 2, 5, 10, 6, 11, 13, 6 "

Initial seq. dictionary :

Index	entry
1	a
2	b
3	r
4	t

" ratatata

→ Dictionary :

Index	Entry
1	a
2	b
3	r
4	t
5	ra
6	at
7	ta
8	ata

LZW decoding

Q- Sequence : " 3, 1, 4, 6, 8, 4, 2, 1, 2, 5, 10, 6, 11, 13, 6 "

Initial seq. dictionary :

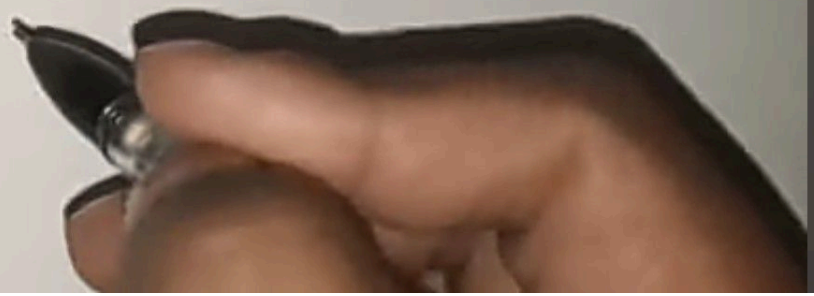
Index	entry
1	a
2	ß
3	r
4	t

" ra t ata ta t ß a ß

ra

→ Dictionary :

Index	Entry
1	a
2	ß
3	r
4	t
5	ra
6	at
7	ta
8	ata



LZW decoding

Q- Sequence : " 3, 1, 4, 6, 8, 4, 2, 1, 2, 5, 10, 6, 11, 13, 6 "

Initial seq. dictionary :

Index	entry
1	a
2	b
3	r
4	t

" ratatatatbab

ra

→ Dictionary :

Index	Entry
1	a
2	b
3	r
4	t
5	ra
6	at
7	ta
8	ata
9	atat
10	tb
11	ba
12	ab
13	br

LZW decoding

Q- Sequence : " 3, 1, 4, 6, 8, 4, 2, 1, 2, 5, 10, 6, 11, 13, 6 "

Initial seq. dictionary :

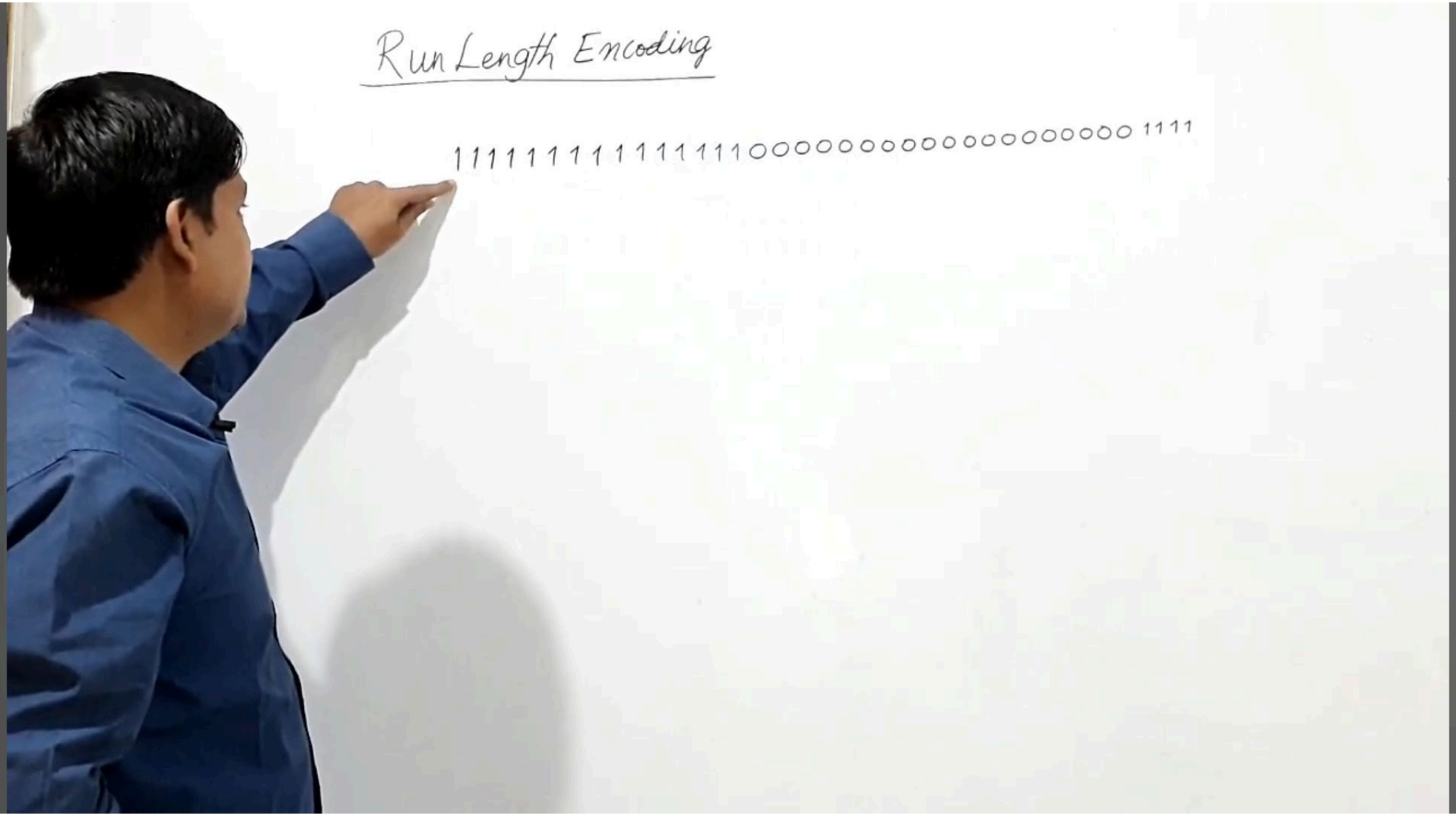
Index	entry
1	a
2	ß
3	r
4	t

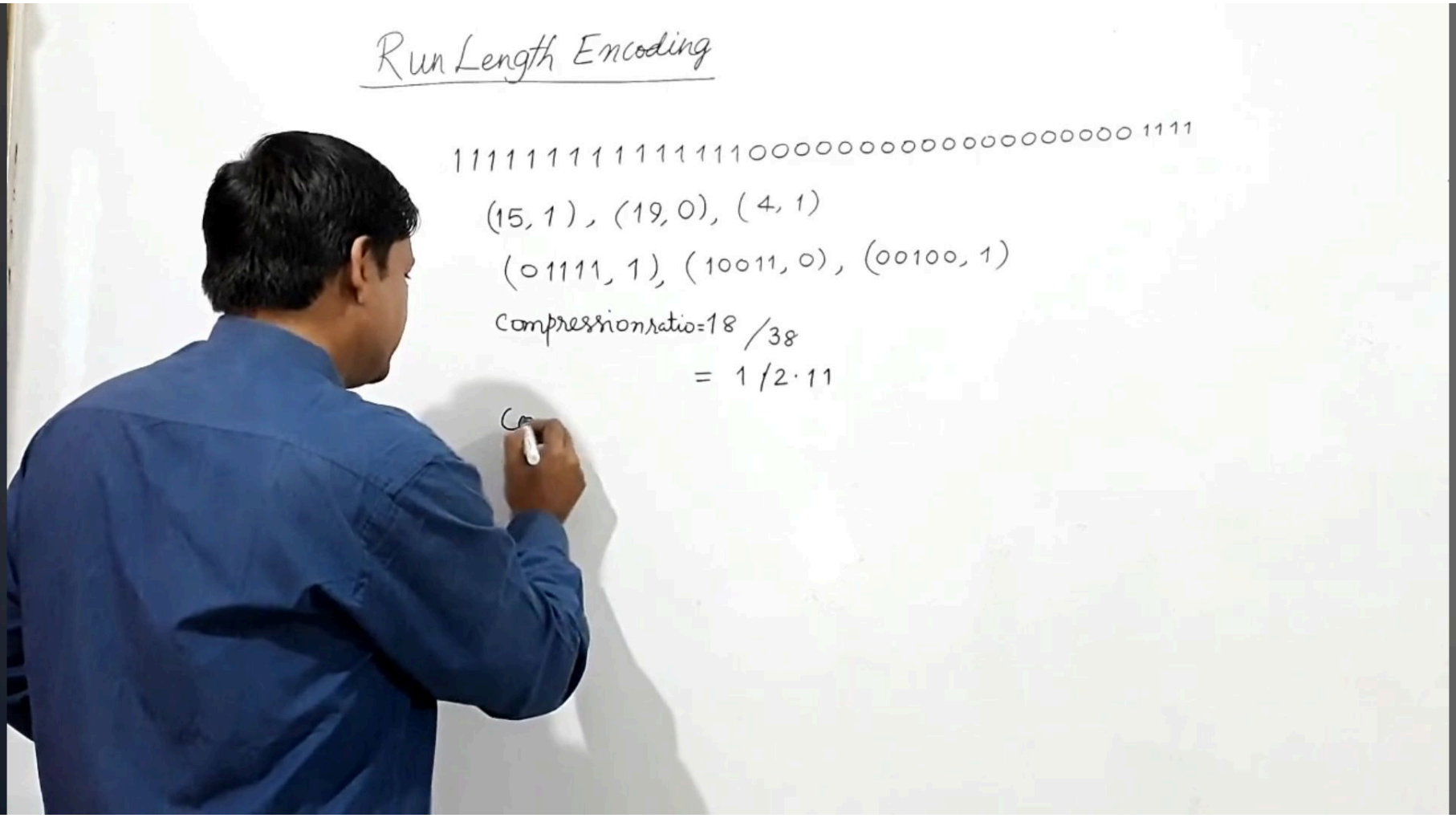
" ratatatatßaß
ratßatßaßrat

→ Dictionary :

Index	Entry
1	a
2	ß
3	r
4	t
5	ra
6	at
7	ta
8	ata
9	atat
10	tß
11	ßa
12	aß
13	ßr

Run length encoding


Run Length Encoding
1111111111111111000000000000000000000 1111


Run Length Encoding
111111111111111000000000000000000 1111
(15, 1), (19, 0), (4, 1)
(01111, 1), (10011, 0), (00100, 1)
Compression ratio = 18/38
= 1/2.11

Run Length Encoding

1111111111111111000000000000000000001111

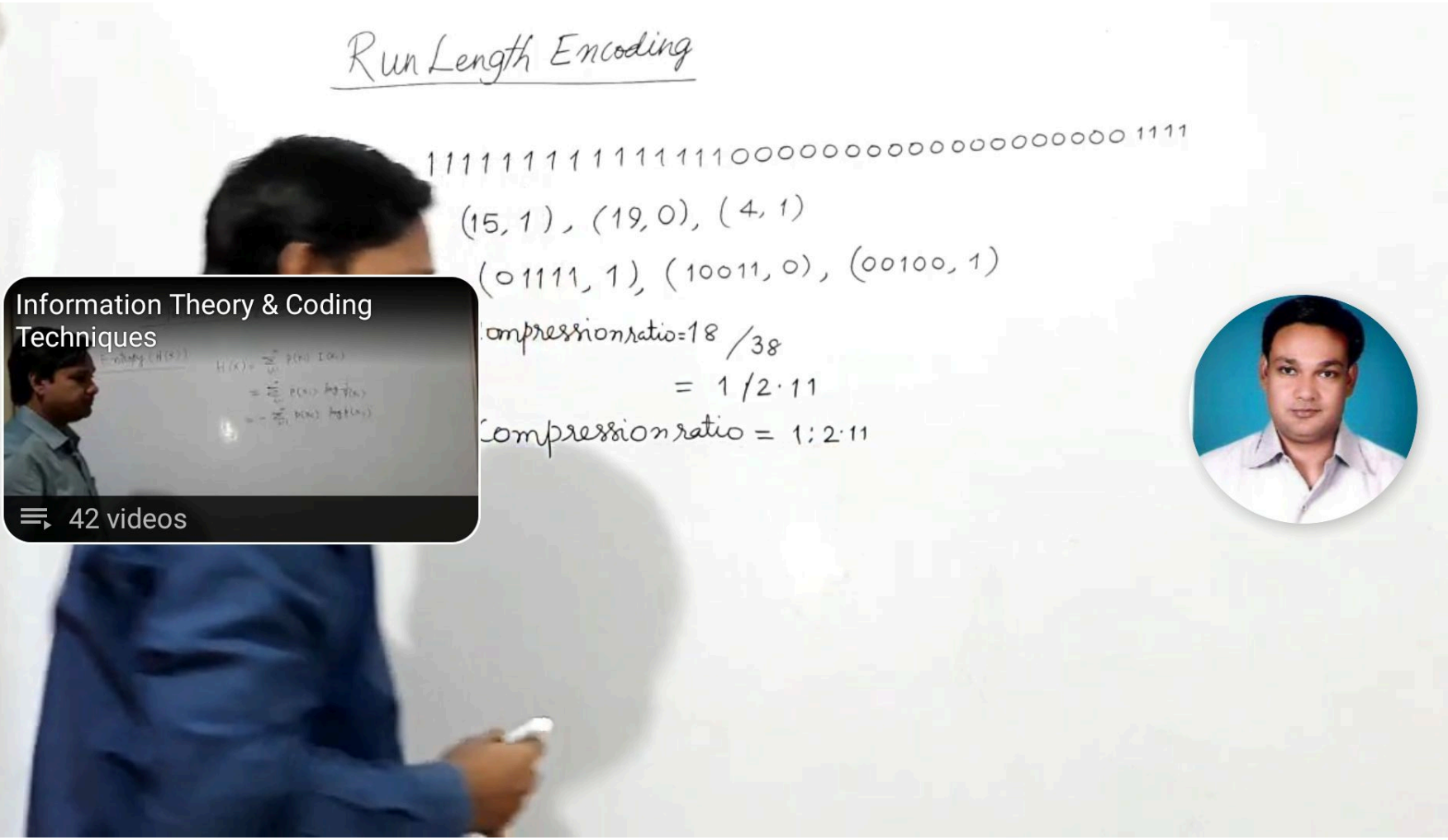
(15, 1), (19, 0), (4, 1)

(01111, 1), (10011, 0), (00100, 1)

Compression ratio = 18 / 38

= 1 / 2.11

Compression ratio = 1 : 2.11



Run length Encoding

⇓

lossless Method

⇓

{text or gr. programs}

⇒ Simplest method of Compression

Example: I/p : AAABBCDDDD

⇓

Encoded : 3A2B1C4D

Run length Encoding

⇓

lossless Method

⇓

{text or gr. programs}

⇒ Simplest method of Compression

Example: I/p : AAABBCDDDD

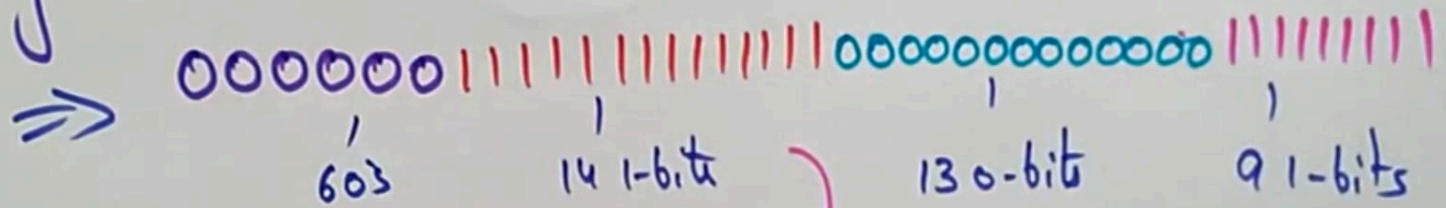
⇓

Encoded : 3A2B1C4D

decoded : AAABBCDDDD

Example!

Original bit stream:



Size ⇒ {42 bits}

① 0:6, 1:14, 0:13, 1:9

② resulting 5-bit bytes

⇒ 00110, 11110, 01101, 11001

⇒ Compressed bit stream

00110111100110111001

⇒ {20 bits}