Data Compression

process of reducing the amount of data required to represent a given quantity of info.

→ 2 GB — 750 mb — compressed.
→ time not → 2hrs — into 1 hr.
→ amount of disk space required.

~~Huffman Code Data Compression~~ -

~~Bas~~

i/p → Encoder —o/p→ Decoder
(uncompressed) ↓ [uncompression]

Data Compression

Code Word length entropy.

Lossless
↓
Text Encryption
↓
exact replica

Lossy.
↓
Some info is lost
↓
less imp. info from the media is removed.
↓
Image, Video, audio
↓
not exact replica

Compression ratio = $\frac{\text{Compressed Size}}{\text{Original Size}}$ $= \frac{750}{2000}$

Compress factor (inverse of cr) = $\frac{1}{\text{Compression ratio}}$

Compression time Time to compress (ms)

Decompression time Time to Decompress (ms)

Fixed length Encoding

a - 97 → 7 bits (1100001)
b :
c : 7×7 = 49 bits
:
g

Compression
3 bits
3×7 = 21 bits

Lossless Compression Technique -
(no loss of info)

Based on binary tree freq-sorting method that allow to encode any msg sequence into shorter encoded msg.
Huffman code is a particular type of optimal prefix code. o/p can be viewed as variable - length code.

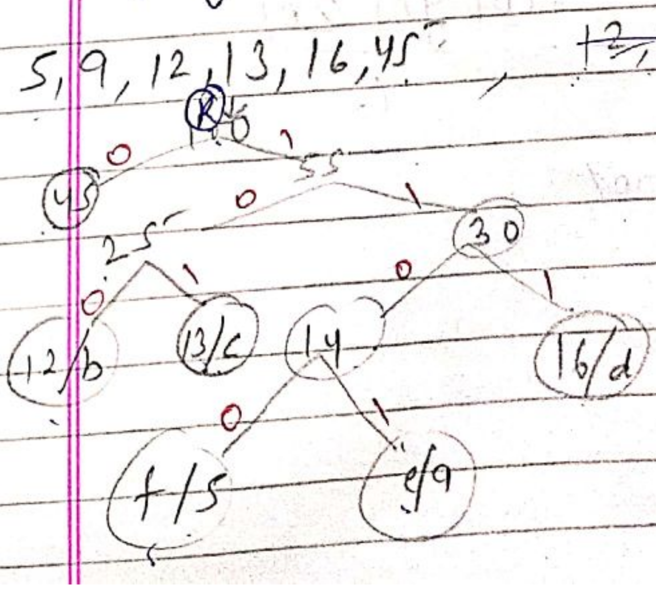
1. Create Sorted nodes on frequency / probability
2. Start
3. find & remove two smallest probability node
4. Create new node, weight[node] = W(A) + W(B)
5. Insert new node, back to Sorted list.
6. Repeat the loop until the list consist of the only last node

Char.	frequency		fixed length code	Var. length code	Code length
a	45		000	0	1
b	13	} 25	001	101	3
c	12		010	100	3
d	16		011	1111	4
e	9	} 14	100	1101	4
f	5		101	1100	4

79

highest Compression
Σ Var. length

5, 9, 12, 13, 16, 45 , ~~12~~, 13, 14, 16, 45 , [~~14~~, 16, 25, 45]
25, 30, 45



lossless

Run length Encoding – Simplest

In this runs of data are stored as a single data value and count rather than original run.

↓ Sequence of same symbol / data value

AAAAAAAA → A (8) count
data value

eg. BBBBBBBBB AAAAA N 44 MMM → B09A05 N01 402 M03

9 5 1 2 3

(20) (15)

but for 0 & 1

00000000000000 | 0000 | | 000000000000

14 4 0 12

↓

1110 0100 0000 1100

Drawback – xyz → x01 y01 z01

3 6

When repeating value not.

PAGE NO.:
DATE: / /

Arithmatic coding

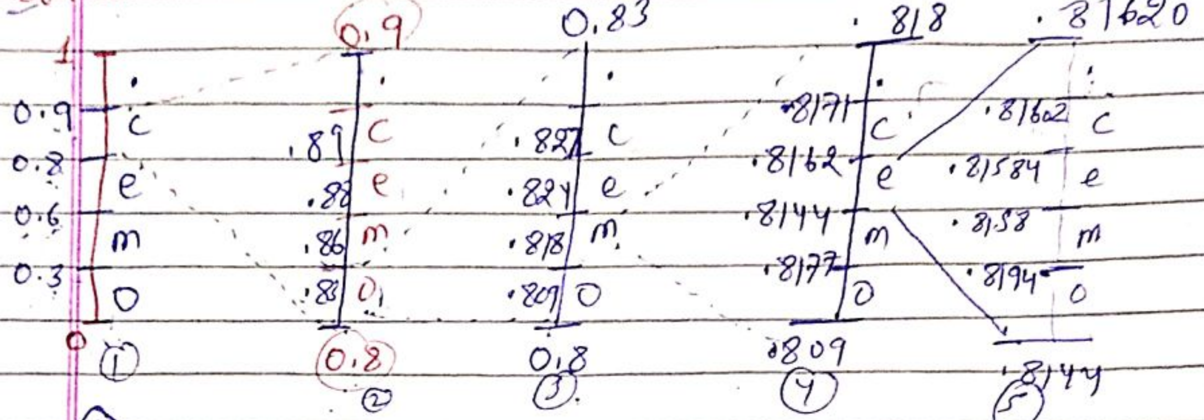
form of entropy encoding used in lossless data c.

- ✓ Efficient code - not depend on size ~~st~~
- ✓ Non block code - block of data on the probability of data
  Stream based Encoding

- ✓ Work very well for sequence $\bar{c}$ low entropy.

o = 0.3   m = 0.3   e = 0.2   c = 0.1   · = ·1

COME



①

② lowe limit + diff (Prob.)

=0.8 + 0.1 (0.3)

o = = 0.8 + ·03 = ·83 = 0

m = ·83 + 0.1 (0.3) = ·86

e = ·86 + 0.1 (0.2) = ·88

c = ·88 + ·1 (0.1) = ·89

③ o = ·8 + ·03 (0.3) = ~~·809~~ ·8 + ·009 = ·809

m = ·809 + ·03 (0.3) = ·818

e = ·818 + ·03 (0.2) = ·824

c = = ·827

⑤

·81602 < Codeword Range < ·81620

Generation Tag = $\frac{L.L + U.L}{2}$ = $\frac{·81602 + ·81620}{2}$

= ·81611.

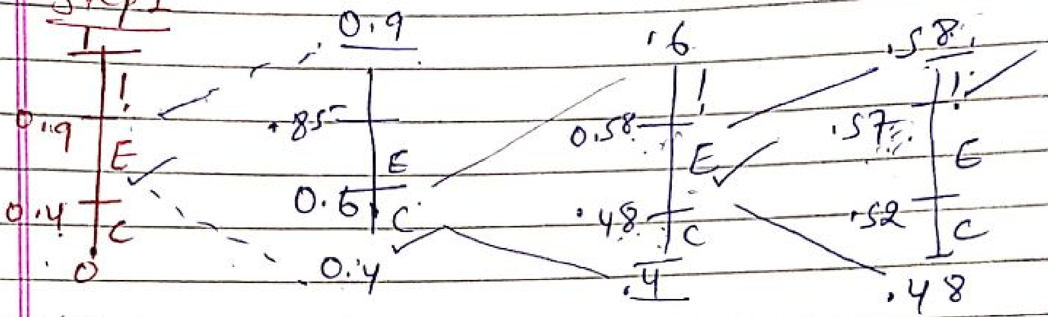
PAGE NO. :
DATE : / /

Decoding

Codeword = 0.572

Symbol	!	C	E
prob	.1	.4	.5

Step 1



based on probability

diff. d = upper bound - lowerbound = 0.9 - 0.4 = 0.5

Range of symbol = ~~lower limit~~ : lower limit + d (.prob. of symbol)

Range of C = 0 ~~.4~~ : 0.4 + 0.5(0.4) = 0.6

E = = .85

.572. is between 0.57 & 0.58

range of termination

ECE!

finite precision Arithmetic