ITC Mod 2

- **Source Encoding** — output of info source → binary sequence

Process by which output of an Info Source is converted into a binary sequence as its output. System doing this: Source Encoding

Code
- Block
  - Non Uniquely Decodable
  - Uniquely (Non-Singular)
    - Instantaneous
      - Optimal
      - Non-Optimal
    - Non Instantaneous
- Non-Block

1 Block Coding: code which maps each symbol of source alphabet 'S' into some 'finite sequence' of code symbols from code alphabet 'X'.
finite sequence → code-word

Source symbol ← $S = \{S_1, S_2, S_3, S_4\}$
↓ to ← $X = \{0, 1\}$: code alphabet
finite sequence

2 Non-Singular: All the code words need to be easily distinguishable

$S_1 S_2 : 000$ $S_2 S_1 : 000$

They are same, so they are singular.

3 Uniquely Decodable Codes :- [non-singular]

Every symbol should be uniquely decodable at receiver.

Example. if $S_1 S_1 : 0000$ $S_1 S_2 : 0001$ $S_2 S_1 : 0100$ and so on, then its non-singular as its unique

4 Instantaneous Codes: [uniquely decodable]

We should be able to recognize the end of the code word sequence by looking at the starting codes

Inst	Inst	Not Inst
00	0	0
01	10	01
10	110	0011
11	1100	0111

Page No.
Date

5 Optimal Codes : Inst. codes with minimum average length L.

- Shannon Fano Encoding Algorithm

$\hat{H} = \frac{1}{N} \sum_{i=1}^{n} P_i \, n_i$ $\qquad R_e = 1 - e$

normally always 1

$H = \sum_{i=1} P_i \log_2 1/P_i$ $\qquad e = H/\hat{H} \times 100\%$

Questions :-

1 Find the code words occurring in the probability 1/2, 1/4, 1/8 and 1/8 for symbols s_1, s_2, s_3, s_4.

Symbols	P_i	I	II	III	C_i	length n_i
s_1	1/2	0			0	1
s_2	1/4	1	1/4 → 10		10	2
s_3	1/8		1/8 → 11	1/8 → 110	110	3
s_4	1/8		1/8	1/8 → 111	111	3

$H = \sum_{i=1}^{n} P_i \log_2 1/P_i = \frac{1}{2}\log 2 + \frac{1}{4}\log 4 + \frac{1}{8}\log 8 \times 2$

$= 1.75$ bits / symbols.

$\hat{H} = \frac{1}{N}\sum_{i=1}^{4} P_i \, n_i = 1.75 \times 1/1 = 1.75$ bits/symbol

$e = H/\hat{H} = 1.75/1.75 = 1$

$e\% = 100\%$

2 Find the code word for the probability 1/4, 1/4, 1/8, 1/8, 1/16, 1/16, 1/16, 1/16 for the symbols $s_1 \ldots s_8$

Symbols	P_i	I	II	III	IV	C_i	n_i
s_1	1/4	0	1/4 → 00			00	2
s_2	1/4		1/4 → 01			01	2
s_3	1/8		1/8 → 10	1/8 → 100		100	3
s_4	1/8		1/8	1/8 → 101		101	3
s_5	1/16	1	1/16	1/16 → 110	1/16 → 1100	1100	4
s_6	1/16		1/16 → 11	1/16	1/16 → 1101	1101	4
s_7	1/16		1/16	1/16 → 111	1/16 → 1110	1110	4
s_8	1/16		1/16	1/16	1/16 → 1111	1111	4

$H = \sum_{i=1}^{8} P_i \log_2 (1/P_i) = [1/4 \log_2 4] \times 2 + [1/8 \log_2 8] \times 2 + [1/16 \times \log 16] \times 4$

$= 2.75$ bits/symbol

$\hat{H} = \sum_{i=1}^{8} 1/N \, P_i \, n_i \quad 1 \times 2.75 = 2.75$ bits/symbols

$e = \frac{H}{\hat{H}} \times 100 = 1 \times 100 = 100\%$

Redundancy $R_e = 1 - e = 0\%$

3 Using Shannon Fano, find code word for P_i : 33/64, 1/4, 1/8, 1/16, 1/32, 1/64 for the symbols $S_1 \ldots S_6$

Symbols	P_i	I	II	III	IV	V	C_i	n_i
S_1	33/64	0					0	1
S_2	1/4	1	1/4 10				10	2
S_3	1/8	1	1/8	1/8 110			110	3
S_4	1/16	1	1/16 11	1/16	1/16 1110		1110	4
S_5	1/32	1	1/32	1/32 111	1/32 1111	1/32 11110	11110	5
S_6	1/64	1	1/64	1/64	1/64	1/64 11111	11111	5

$H = \sum_{i=1}^{6} P_i \log_2 (1/P_i) = 33/64 \log 64/33 + 1/4 \log 4 + 1/8 \log 8$

$+ 1/16 \log 16 + 1/32 \log 32 + 1/64 \log 64$

$= 1.867$ bits/symbols

$\hat{H} = 1/N \sum_{i=1}^{6} P_i \, n_i = 1 \times 1.867 = 1.867$ bits/symbols

$e = H/\hat{H} \times 100 = 1 \times 100 = 100\%$.

- Huffman Encoding

Decreasing order of P_i

$r = 2 \quad (x = 0, 1)$: Binary Huffman Coding

$r = 3 \quad (x = 0, 1, 2)$: Ternary

$r = 4 \quad (x = 0, 1, 2, 3)$: Quarternary

$e = \frac{H}{\bar{L} \log_2 r} \qquad \bar{L} = \sum_{i=1}^{n} P_i n_i \qquad n = r + \alpha (r-1)$

$\sigma^2 = \sum_{i=1}^{n} P_i (n_i - \bar{L})^2$

Questions :-

1 Find Huffman codes, efficiency and variance of 0.4, 0.2, 0.2, 0.1, 0.1

S_i	P_i	I	II	III	Code	n_i
S_1	0.4	0.4	0.4	0.6]^0	00	2
S_2	0.2	0.2	0.4]^0	0.4]$_1$	10	2
S_3	0.2	0.2]^0	0.2]$_1$		11	2
S_4	0.1]^0	0.2]$_1$			010	3
S_5	0.1]$_1$				011	3

$n = r + \alpha(r-1) \rightarrow 5 = 2 + \alpha(2-1) \rightarrow \alpha = 3$

$H = \sum_{i=1}^{5} P_i \log_2 1/P_i = 0.4 \log \frac{1}{0.4} + 0.2 \log \frac{1}{0.2} \times 2 + 0.1 \log \frac{1}{0.1} \times 2$

$= 2.1216$ bits / symbols

$\bar{L} = \sum_{i=1}^{n} P_i n_i = 0.4 \times 2 + 0.2 \times 2 + 0.2 \times 2 + 0.1 \times 3 + 0.1 \times 3$

$= 2.2$ bits / symbol

$e = \frac{H}{\bar{L} \log_2 r} = \frac{2.1216}{2.2 \log_2 2} = 96.4\%$

$\sigma^2 = \sum_{i=1}^{n} P_i (n_i - \bar{L})^2 = 0.4(2-2.2)^2 + 0.2 \times 2 \times (2-2.2)^2 + 0.1(3-2.2)^2 \times 2$

Variance

$\sigma = 0.16$

2 Binary Huffman Coding of 1/2, 1/4, 1/8, 1/8

S_i	P_i	I	II	Code	n_i
S_1	1/2	1/2	1/2]^0	1	1
S_2	1/4	1/4]^0	1/2]$_1$	01	2
S_3	1/8]^0	1/4]$_1$		000	3
S_4	1/8]$_1$			001	3

$H = 0.5 + 0.5 + 0.75 = 1.75$ bits / symbols

$\bar{L} = 1.75$ bits / symbols

$e = \frac{H}{\bar{L} \log_2 r} = \frac{1}{1} = 100\%$

3 Quartonary coding of 0.11 × 9 times

→ $n = r + \alpha(r-1) \rightarrow 9 = 4 + \alpha(4-1)$

$\alpha = 1.6$ ∴ It should be integer.

Add dummy prob = 0

$n = r + \alpha(r-1) \rightarrow 10 = 4 + \alpha \times 3 \rightarrow \alpha = 2$

S_i	P_i	I	II	C_i	n_i
S_1	0.11	0.33	0.44	2	1
S_2	0.11	0.11	0.33	3	1
S_3	0.11	0.11	0.11	00	2
S_4	0.11	0.11	0.11	01	2
S_5	0.11	0.11		02	2
S_6	0.11	0.11		03	2
S_7	0.11	0.11		10	2
S_8	0.11			11	2
S_9	0.11			12	2
S_{10}	0.0			13	2

• LZW Encoding

1 Sequence ababbabcababba

dictionary

index	entry
1	a
2	b
3	c

Initial LZW dictionary :-

Encoded o/p	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	–	–

Output : 1, 2, 4, 5, 2, 3, 4, 6, 1

Page No.
Date

- Run Length Encoding : lossless compression where we reduce bits by assigning different values to the code

1 $\underbrace{00000}_{5}$ $\underbrace{11111}_{5}$ $\underbrace{00}_{2}$ $\underbrace{1}_{1}$ $\underbrace{0000}_{4}$ $\underbrace{1}_{1}$ $\underbrace{0}_{1}$ $\underbrace{1}_{1}$

→ (0,5) , (1,5) , (0,2), (1,1) , (0,4), (1,1), (0,1), (1,1)

n = 8 $\log_2 n$ = 3 bits , the code can be only 3 bits long

(0,101), (1,101), (0,010), (1,001), (0,100), (1,001), (0,001) , (1,001)

Concatenate : 0101110100101001 0100 1001 00011001

2 $\underbrace{00000}_{5}$ $\underbrace{111111}_{6}$ $\underbrace{00000}_{5}$ $\underbrace{111111}_{6}$ $\underbrace{0}_{1}$ $\underbrace{11}_{2}$ $\underbrace{00}_{2}$ $\underbrace{1111}_{4}$ $\underbrace{00}_{2}$

→ (0,5), (1,6), (0,5), (1,6), (0,1), (1,2), (0,2), (1,4), (0,2)

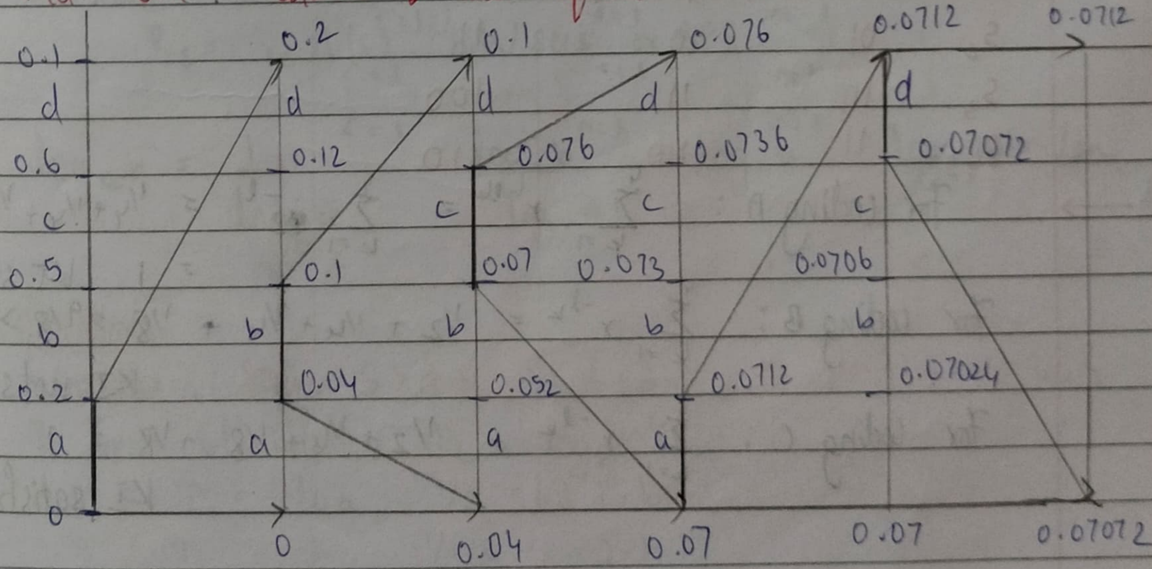
n = 9 $\log_2 9$ = 3log3 = 4.754

(0,0101), (1,0110), (0,0101), (1,0110), (0,0001), (1,0010), (0,0010), (1,0101), (0,0010)

00101101100010110110000011001000001010100010

- Arithmetic Coding

1 Consider the probabilities : P(a) = 0.2 P(b) = 0.3 P(c) = 0.1 P(d) = 0.4 . Encode the sequence abcad.



d_i for a = upper limit - lower limit = 0.2 - 0 = 0.2
Range of symbol = LL : LL + d_i (Prob of sym)

Range of 'a' = 0 : 0 + 0.2 (0.2) = 0 : 0.04
Range of 'b' = 0.04 : 0.04 + 0.2 (0.3) = 0.04 : 0.1
Range of 'c' = 0.1 : 0.1 + 0.2 (0.1) = 0.1 : 0.12
Range of 'd' = 0.12 : 0.12 + 0.2 (0.4) = 0.12 : 0.2

d_i of b = UL - LL = 0.1 - 0.04 = 0.06
Range of 'a' = 0.04 : 0.04 + 0.06 (0.2) = 0.04 : 0.052
and so on :-

$$\text{Tag} = \frac{UL + LL}{2} = \frac{0.0712 + 0.07072}{2}$$

= 0.0706 : Encoded message.

- Kraft's Inequality :-

$\sum_{k=1}^{n} r^{-l_k}$ $r = 2$

= 1 satisfied
< 1 satisfied
> 1 not satisfied

l_k : length of k

1

	Code A	code B	code C
S_1	00 $l_k = 2$	0 $l = 1$	0 $l = 1$
S_2	01 $l_k = 2$	10 $l = 2$	11 $l = 2$
S_3	10 $l = 2$	11 $l = 2$	100 $l = 3$
S_4	11 $l = 2$	110 $l = 3$	110 $l = 3$

→ For coding A : $\sum_{k=1}^{4} r^{-l_k} = \sum_{k=1}^{4} 2^{-l} = 1/4 + 1/4 + 1/4 + 1/4$
= 1 KI satisfied

For coding B : $\sum_{k=1}^{4} r^{-l_k} = 1/2 + 1/4 + 1/4 + 1/8 = 9/8 > 1$
KI not satisfied

For coding C : $\sum_{k=1}^{4} r^{-l_k} = 1/2 + 1/4 + 1/8 + 1/8 = 1$
KI satisfied.