- Apply Shannon's binary encoding procedure to the following set of messages and obtain code efficiency and redundancy. 1/8, 1/16, 3/16, 1/4, 3/8
- Solution:
- 3/8>1/4>3/16>1/8>1/8

$\alpha_1 = 0$

$\alpha_2 = P_1 = 3/8 = 0.375$

$\alpha_3 = P_2 + \alpha_2 = 1/4 + 3/8 = 0.625$

$\alpha_4 = P_3 + \alpha_3 = 3/16 + 0.625 = 0.8125$

$\alpha_5 = P_4 + \alpha_4 = 1/8 + 0.8125 = 0.9375$

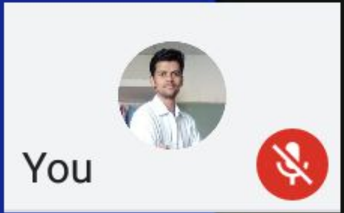
$0.375 \times 2 = 0.75 \rightarrow 0$
$0.75 \times 2 = 1.5 \rightarrow 1$
$0.5 \times 2 = 1$
$\rightarrow \boxed{0.01}$

$0.625 \times 2 = 1.25 \rightarrow 1$
$0.25 \times 2 = 0.50 \rightarrow 0$
1
$\boxed{0.101}$

$0.8125 \times 2 = 1.625 \rightarrow 1$
$0.625 \times 2 = 1.25 \rightarrow 1$
0.25×2
$0.5 \times 2 \rightarrow 0$
1
$\rightarrow \boxed{0.110}$

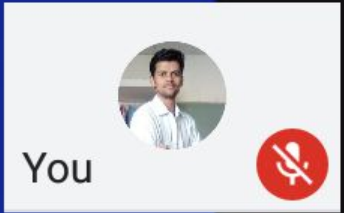
$0.9375 \times 2 = 1.875 \rightarrow 1$
$0.875 \times 2 = 1.75 \rightarrow 1$
$0.75 \times 2 = 1.5 \rightarrow 1$
$0.5 \times 2 = 1$
$\rightarrow \boxed{0.1111}$

- l^i values are: 2,2,3,3,4



S_i	P_i	α_i	l_i	binary	Code
S_1	3/8	0	2	0.00	00
S_2	1/4	0.375	2	0.01	01
S_3	3/16	0.625	3	0.101	101
S_4	1/8	0.8125	3	0.110	110
S_5	1/16	0.9375	4	0.1111	1111

- $H(S) = \frac{1}{4}log_2 4 + \frac{3}{8}log_2\frac{8}{3} + \frac{1}{8}log_2 8 + \frac{3}{16}log_2\frac{16}{3} + \frac{1}{4}log_2 4 + \frac{1}{16}log_2 16$
- $H(S) = 2.1085\ bits/symbol$
- $L = \sum_{i=1}^{q} P_i l_i = \frac{1}{4}(2) + \frac{3}{8}(2) + \frac{1}{8}(3) + \frac{3}{16}(3) + \frac{1}{16}(4)$
- $L = 2.4375\ bits/$symbol

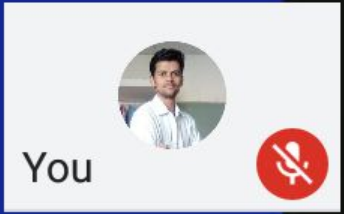


- The average length of this code is
  $L = \sum_{i=1}^{q} P_i l_i$

$$L = 2 \times 0.4 + 2 \times 0.3 + 3 \times 0.2 + 3 \times 0.1 = 2.4 \text{ Binits / message}$$

$$H(S) = 0.4 \log \frac{1}{0.4} + 0.3 \log \frac{1}{0.3} + 0.2 \log \frac{1}{0.2} + 0.1 \log \frac{1}{0.1} = 1.84644 \text{ bits / message;}$$

- $\% \eta_c = \frac{\mathrm{H(S)}}{\mathrm{L}} * 100 = \frac{1.8464}{2.4} * 100 = 76.93\%$

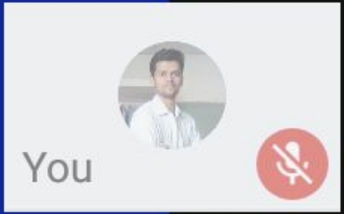


- $\%\eta = \frac{H(S)}{L} * 100 = 86.5\%$
- Redundancy=1- η=100-86.5=13.5%

- Homework: Repeat the above messages $(x_{1,} x_{2,} x_3)$ with P= (1/2, 1/5, 3/10)

Huffman Coding

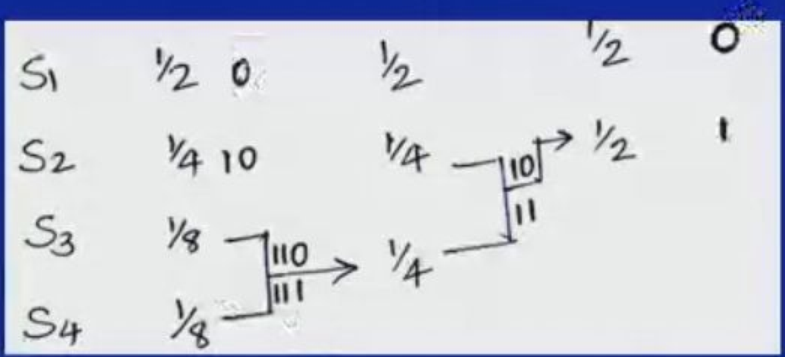
- The source symbols are listed in the decreasing order of probabilities.
- Check if $q = r + a(r-1)$ is satisfied and find the integer 'a', q is number of source symbols and r is number of symbols used in code alphabets.
- If 'a' is not integer, add dummy symbols of zero probability of occurrence.
- Combine the last 'r' symbols into a single composite symbol by adding their probability to get a reduced source.
- Repeat the above three steps, until in the final step exactly ***r***- symbols are left.



- The last source with ‘r’ symbols are encoded with ‘r’ different codes 0,1,2,3,….r-1
- In binary coding the last source are encoded with 0 and 1
- As we pass from source to source working backward, decomposition of one code word each time is done in order to form 2 new code words.
- This procedure is repeated till we assign the code words to all the source symbols of alphabet of source ‘s’ discarding the dummy symbols.

You

- Cont



Symbols	Codes	Probabilities	Length
S_1	0	½	1
S_2	10	¼	2
S_3	110	1/8	3
S_4	111	1/8	3

- Ex.2: A source has 9 symbols and each occur with a probability of 1/9. Construct a binary Huffman code. Find efficiency and redundancy of coding.
- Solution:
- $q = r + \alpha(r - 1)$
- $9 = 2 + \alpha(1)$ => $\alpha = 7 \in Z$